

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

META PLATFORMS, INC.; INSTAGRAM, INC.; WHATSAPP LLC; META
PLATFORMS TECHNOLOGIES, LLC; AND GIPHY, INC.

Petitioners

v.

VL COLLECTIVE IP LLC

Patent Owner

U.S. Patent No. 7,266,682

Case IPR2023-00923

**SUPPLEMENTAL DECLARATION OF DR. HENRY HOUH, PH.D. IN
SUPPORT OF PETITIONERS' REPLY**

UPDATED LIST OF EXHIBITS

Ex. No.	Description
1001	U.S. Patent No. 7,266,682 (“the ’682 Patent” or “the Patent”)
1002	Certified File History of U.S. Patent No. 7,266,682
1003	Declaration of Dr. Henry Houh, Ph.D. in Support of Petition for <i>Inter Partes</i> Review of U.S. Patent No. 7,266,682
1004	Curriculum Vitae of Dr. Henry Houh
1005	M. Handley, J. Crowcroft, & C. Bormann, Very Large Conferences on the Internet: The Internet Multimedia Conferencing Architecture and the MBONE by, https://www.cl.cam.ac.uk/~jac22/otalks/cnisdn-article.pdf (“Handley”)
1006	Erol Basturk, Mihir Bellare, Chee-Seng Chow, & Roch Guérin, Secure Transport Protocols for High-Speed Networks, https://cseweb.ucsd.edu/~mihir/papers/srtp.pdf (“Basturk”)
1007	PacketCable 1.0 Architecture Framework Technical Report, https://community.cablelabs.com/wiki/plugins/servlet/cablelabs/alfresco/download?id=bddae471-01dd-43c8-8e59-d6547fcf4ffd (“PacketCable”)
1008	RTP: A Transport Protocol for Real-Time Application, IETF RFC 1889, https://datatracker.ietf.org/doc/html/rfc1889
1009	Andrew S. Tanenbaum, Computer Networks (3 rd ed. 1996) (“Tanenbaum”)
1010	Declaration of Sylvia D. Hall-Ellis, Ph.D. in Support of Petition for <i>Inter Partes</i> Review of U.S. Patent No. 7,266,682
1011	Reserved
1012	Complaint, <i>VideoLabs, Inc., et al. v. Meta Platforms, Inc. et al.</i> , Civil Action No. 1-22-cv-00680 (D. Del.)
1013	Internet Protocol – RFC 791, DARPA Internet Program (September 1981)
1014	Carrier Sense Multiple Access with Collision Detection, IEEE Standard 802.3, IEEE Standards for Local Area Networks (1985)

1015	David D. Clark & David L. Tennenhouse, Architectural Considerations for a New Generation of Protocols, Laboratory for Computer Science, M.I.T. (1990), https://dl.acm.org/doi/10.1145/99517.99553 .
1016	Declaration of Dr. Jon Crowcroft
1017	Supplemental Declaration of Dr. Henry Houh
1018	Supplemental Declaration of Dr. Sylvia Hall-Ellis relating to Handley (EX 1005)
1019	Supplemental Declaration of Dr. Sylvia Hall-Ellis relating to PacketCable (EX 1007)
1020	Declaration of Nathaniel E. Frank-White
1021	Web Archive of UCL Web Servers Search (August 16, 2000), available at https://web.archive.org/web/20000816095229/http://www.ucl.ac.uk/search/compass/
1022	Web Archive of Mark Handley's List of Papers, Publications, and Specifications (February 10, 1999) available at https://web.archive.org/web/19990210102149/http://north.east.isi.edu/~mjh/papers.html
1023	Web Archive of Mark Handley's University of Southern California Information Sciences Institute Website (January 30, 1998), available at https://web.archive.org/web/19980130215743/http://north.east.isi.edu/~mjh/
1024	Web Archive of Dr. Jon Crowcroft's List of Papers (August 28, 1999), available at https://web.archive.org/web/19990828053320/http://www.cs.ucl.ac.uk/staff/jon/jons-papers.html
1025	Web Archive of UCL's DARPA FTP Server (February 17, 2001), available at https://web.archive.org/web/20010217183731/http://www.cs.ucl.ac.uk/darpa
1026	Web Archive of Dr. Jon Crowcroft's UCL Website (December 3, 1998), available at

	https://web.archive.org/web/19981203112130/http://www.cs.ucl.ac.uk:80/staff/jon/
1027	M. Handley, J. Crowcroft, C. Bormann, & J. Ott, <i>Very Large conferences on the Internet: the Internet multimedia conferencing architecture</i> , 31 COMPUT. NETWORKS 191 (1999), https://www.sciencedirect.com/science/article/abs/pii/S0169755298002621#preview-section-abstract (last visited Sep 25, 2023).
1028	M. Handley, J. Crowcroft, C. Bormann, & J. Ott, <i>Very Large conferences on the Internet: the Internet multimedia conferencing architecture</i> , 31 COMPUT. NETWORKS, 191 (1999).
1029	Cross-Examination Transcript of Dr. Henry Houh dated February 16, 2024
1030	Declaration of Ethan Perbohner
1031	Cross-Examination Transcript of Dr. Stuart Stubblebine dated May 15, 2024
1032	IETF RFC 1936
1033	Supplemental Declaration Of Dr. Henry Houh, Ph.D. In Support Of Petitioners' Reply
1034	U.S. Patent No. 6,526,056

Unless otherwise noted, all emphasis added.

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	QUALIFICATIONS	2
III.	CLAIM CONSTRUCTION	3
IV.	A POSITA WOULD HAVE BEEN MOTIVATED TO USE BASTURK’S SECURE TRANSPORT PROTOCOL WITH HANDLEY’S INTERNET CONFERENCING ARCHITECTURE (GROUND I)	3
A.	Basturk’s “On-The-Fly” Means MAC Generation Occurs as Data Transmissions Proceed	3
B.	Basturk’s Teaching of “On-The-Fly” Would Motivate a POSITA to Place a Signature (MAC) at the End of a Packet	7
C.	Dr. Stubblebine’s Understanding of OSI Layers is Overly Strict.....	11
D.	Basturk’s Secure Transport Protocol Is Not A Reliable Delivery Protocol	16
E.	Basturk’s Secure Transport Protocol and Authentication Techniques are Compatible with Handley’s Internet Conferencing Architecture	20
F.	Element [1.1]	26
G.	Element [1.2]	29
H.	Element [1.3]	33
I.	Element [1.4]	34
J.	Independent Claims 25-28	34
K.	Dependent Claims 2-21, and 23.	34
V.	GROUND II.....	35
A.	Element [1.1]	35

B.	Element [1.2]	36
C.	Element [1.3]	47
D.	Element [1.4]	47
E.	Independent Claims 25-28	47
F.	Dependent Claims 22 and 24	47
G.	Dependent Claims 2-21 and 23	48
VI.	GROUND III	48
A.	Motivation to Combine	48
B.	Independent Claim 1	49
C.	Independent Claims 25-28	50
D.	Dependent Claims 22 and 24	51
E.	Dependent Claims 2-21 and 23	51
VII.	GROUND IV	51

1. I, Henry Houh, declare as follows:

I. INTRODUCTION

2. I am over the age of eighteen (18) and otherwise competent to make this Declaration.

3. I have been retained by Meta Platforms, Inc, Instagram, Inc., WhatsApp LLC, Meta Platforms Technologies, LLC, and GIPHY, Inc. (“Petitioners”) as an independent, technical expert consultant in connection with the above-captioned Petition for *Inter Partes* Review (“IPR”) of U.S. Patent No. 7,266,682 (“the ’682 Patent”) (EX1001). I am being compensated at my rate of \$680 per hour for my services. No part of my compensation is dependent on my opinions or on the outcome of this proceeding. I have no financial interest in any of the parties in this proceeding.

4. I have been asked to prepare this Supplemental Declaration (“Supplemental Declaration”) in Support of Petitioners’ Reply in order to provide my opinions regarding the statements made by Patent Owner and the opinions expressed in the Declaration of Stuart G. Stubblebine, Ph.D., EX2004 (“Stubblebine Declaration”). I previously provided my opinions in support of Petitioners’ Petition for *Inter Partes* Review, EX1003 (“First Declaration”).

5. In forming my opinions expressed in this Supplemental Declaration, I relied upon my education and experience, which I describe in Section II of my First

Declaration and in my *curriculum vitae*, EX1004. I considered the viewpoint of a person having ordinary skill in the art (“POSITA”), which I define in Section IV of EX1003, as of the priority date of the ’682 Patent. My opinions are based, at least in part, on the Exhibits attached to my First Declaration, the Stubblebine Declaration and Exhibits referenced therein, and the Exhibits attached to VideoLabs’ Replies, e.g., EX2002 and EX2003. I also considered the ’682 Patent and its file history in forming my opinions.

6. For the purposes of this IPR, I have been instructed to assume that the effective filing date of the Claims of the ’682 Patent challenged by the Petitioners in this IPR is no earlier than January 18, 2000.

7. My analysis of the materials relating to this matter is ongoing and I will continue to review any new material as it is provided. I reserve the right to revise, supplement, and/or amend my opinions stated in this Declaration based on new information and my continuing analysis of the materials.

II. QUALIFICATIONS

8. As I stated in my First Declaration, in my opinion, a POSITA would have had a Bachelor’s degree in computer science, electrical engineering, or a related field, and two years of experience; or by obtaining a Master’s degree in electrical engineering or a related field, but having only one to two years of experience; or by having no formal education but experience in network data transmission and

computer networking of at least eight years. Additional education might compensate for less experience, and vice-versa.

9. I understand Dr. Stubblebine has adopted my level of ordinary skill and has not put forth his own definition of a POSITA. As of January 2000, I exceeded the qualifications for POSITA based on my academic achievement and employment experience. *See* my qualifications and professional background in Section II of my First Declaration and in my *curriculum vitae*, EX1004.

III. CLAIM CONSTRUCTION

10. Dr. Stubblebine sets forth a claim construction for two terms, “application layer,” and “application protocol.” POR, 14; EX2004, ¶60. Even under PO’s proposed claim constructions, it does not render any asserted reference inapplicable, nor does it change that those references either disclose and/or render obvious all Challenged Claims.

IV. A POSITA WOULD HAVE BEEN MOTIVATED TO USE BASTURK’S SECURE TRANSPORT PROTOCOL WITH HANDLEY’S INTERNET CONFERENCING ARCHITECTURE (Ground I)

A. Basturk’s “On-The-Fly” Means MAC Generation Occurs as Data Transmissions Proceed

11. As noted in my First Declaration, Basturk states “[i]deally, we want signatures to be computed and verified on-the-fly; i.e., MAC generation should be carried out as data transmissions proceeds rather than before these operations have completed.” EX1003, ¶84 (citing EX1006, 12 (cleaned up)).

12. Dr. Stubblebine opines that when Basturk uses “on-the-fly,” “Basturk is referring to starting the process of computing the style of MAC during (or in parallel with) the process of *retrieving context data* for performing cryptographic operations.” EX2004, ¶107 (emphasis in original). I disagree. Basturk is clear about what it refers to when it uses the term “on-the-fly.” In full, Basturk states that “[i]deally, we want signatures to be computed and verified on-the-fly; i.e., MAC generation (verification) should be carried out as data transmissions (receptions) proceeds rather than before (after) these operations have completed.” EX1006 at 12. A POSITA would have understood Basturk’s statement as a shorthand for two separate statements: (1) “[i]deally, we want signatures to be **computed on-the-fly**; i.e., MAC **generation** should be carried out as data **transmissions** proceeds rather than **before** these operations have completed” and (2) “[i]deally, we want signatures to be **verified on-the-fly**; i.e., MAC **verification** should be carried out as data **receptions** proceeds rather than **after** these operations have completed.” EX1006, 12. A POSITA would have understood these statements to mean that “on-the-fly” during data transmission is the generation of the MAC is carried out as data transmissions proceeds, i.e., in parallel, and MAC verification is carried out as data receptions proceeds.

13. Throughout his analysis, Dr. Stubblebine routinely overlooks and ignores Basturk’s teachings and limits his “on-the-fly” analysis to data receptions

(where context information must be retrieved), excluding data transmission. EX2004, ¶¶84-88. Dr. Stubblebine incorrectly pulls from Basturk’s teachings that “[t]he ability to achieve such on-the-fly processing depends on two major components. The first is the latency in retrieving the information needed to perform the necessary computations,” and “the second component for achieving on-the-fly processing ‘is the nature of the computations themselves.’” EX2004, ¶¶85-86 (quoting EX1006 at 12-13). However, Dr. Stubblebine ignores the direct teaching in Basturk that recites “[t]he first component is most relevant at the receiver, while the second [MAC computations] is equally important at **both** the transmitter and the receiver.” EX1006, 13. Dr. Stubblebine addresses the style of computation in his report only in the receive context, ignoring the transmission context which Basturk teaches is “equally important.” *Id.*; EX2004, ¶88. Unlike how a POSITA would have understood Basturk, none of Dr. Stubblebine’s opinions directly address the transmission context of “on-the-fly.”

14. Dr. Stubblebine’s understanding of “on-the-fly” does not conform whatsoever with the teachings of Basturk. For example, and with respect to data transmission, Basturk is clear that “computed on-the-fly” means that the generation of the MAC occurs as data transmissions proceed and that “these operations” refer to the operations for data transmission, not the process of retrieving context data as Dr. Stubblebine asserts. *Id.*; EX2004, ¶¶106-107. A POSITA would have understood

that generation of the MAC as data transmission proceeds matches Basturk's teaching that "[i]n all instances we would like all computations to be carried out 'on-the-fly' as data is being transmitted or received. This minimizes latency." EX1006, 7. In contrast, Dr. Stubblebine's retrieval context interpretation of "on-the-fly" does not make sense for data transmission. EX2004, ¶¶106-107. Where connection retrieval is mentioned in Basturk, it states that:

Both signature computation and encryption/decryption rely on the use of secret information (key) shared by the two communicating parties. Retrieval of this information is, therefore, necessary for such processing. While one can assume *its immediate availability when transmitting data*, it must typically be retrieved upon receipt of a packet and this context retrieval step can introduce latency.

EX1006 at 8. As seen, Basturk teaches that the latency due to retrieving the context information is upon receipt of a packet, not upon transmission of a packet, where the context information is "immediately availab[le]." *Id.* Because Basturk teaches context information is immediately available during data transmission—meaning there is no latency—Basturk's "on-the-fly" technique and goal of "minimize[ing] latency" would provide no benefit during data transmission, as there is no latency to minimize. EX1006, 7-8. In contrast, a POSITA would have understood that to minimize latency during data transmission, Basturk's "on-the-fly" teaching means that "MAC generation should be carried out as data transmission proceeds," to reduce latency introduced by MAC computations. EX1006, 7.

B. Basturk’s Teaching of “On-The-Fly” Would Motivate a POSITA to Place a Signature (MAC) at the End of a Packet

15. As detailed in my First Declaration and above, “on-the-fly” refers to, in part, the process of computing the MAC in parallel to the transmission of the packet, and then inserting the MAC at the end of a packet. EX1003, ¶¶65, 85. Dr. Stubblebine opines that “‘on-the-fly’ processing does not require or motivate placing the signature at the end of the packet.” EX2004 at ¶106. I disagree. Basturk states “[t]he signature covers not only the data payload but also the transport header.” EX1006 at 9. A POSITA would have understood that when the MAC is computed “on-the-fly” over the data payload and transport header as data transmission of the packet proceeds, this can only be done if the MAC is placed at the end of a packet. If the MAC were placed somewhere other than at the end of the packet, the “on-the-fly” MAC computation during data transmission cannot be computed for the data in the packet that is transmitted after the fully computed MAC is inserted into the packet. Were the MAC located at the beginning or in the middle of the packet, the “on-the-fly” computed MAC would not be computed over the data transmitted after the MAC is inserted and, thus, cannot accurately reflect the correct MAC computed over the whole packet.

16. Further, and as described in my First Declaration, the “Implementation Overview” Section of Basturk describes placing the MAC at the end of the whole packet.

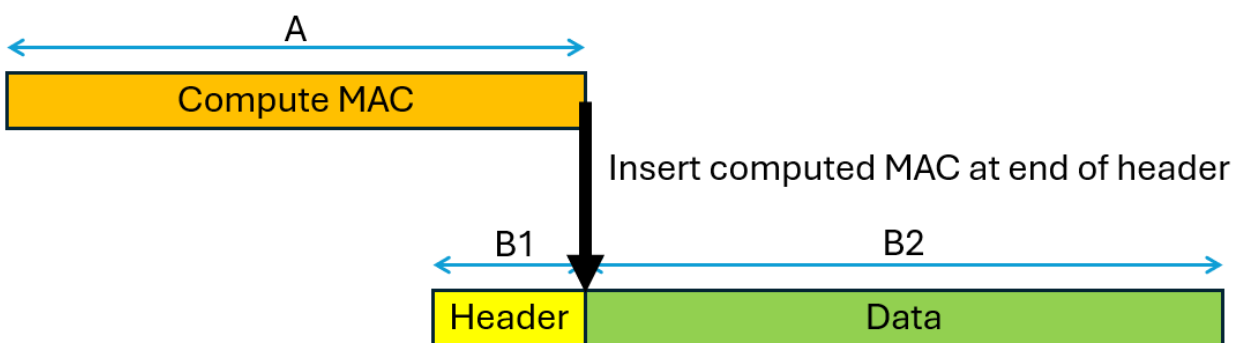
A transport packet consists of a header followed by data, if any. The format of the packets has been designed to be extensible. Most transport control functions are present in optional fields called segments which are part of the transport header. A security segment has been added to include information needed for authentication and confidentiality. ***Following data, a trailer has been added that contains the packet signature.***

EX1006, 16; EX1003, ¶85. A POSITA would have understood and found obvious that the “trailer” is data that can be added to the end of the packet. This trailer, as described in Basturk can contain the authentication data, i.e., signature or MAC.

17. Dr. Stubblebine incorrectly provides two reasons why a POSITA would not be motivated to place a signature at the end of a packet. First, Dr. Stubblebine wrongly states that “a POSITA would have understood that even with Basturk’s ‘on-the-fly’ processing, packets would not be passed to the next layer until all computation for that packet are complete.” EX2004, ¶108. I disagree and address this later in full in Section IV.C of this Declaration. *See* § IV.C. Second, Dr. Stubblebine wrongly opines that there are already methods in place where the MAC can be placed in the RTP packet header. EX2004, ¶134.

18. If the MAC must be fully computed before the packet can begin to be transmitted, as Dr. Stubblebine opines, a POSITA would have understood there is no advantage to placing the MAC in the packet header versus at the end of a packet. This is because the packet cannot be transmitted until the MAC computation is complete. Further, when computing the MAC “on-the-fly,” placing the MAC in the

packet header results in a longer time delay to when the packet is fully transmitted (measured from the time the MAC is computed) than placing the MAC at the end of a packet. If the MAC is placed in the packet header (for example, at the end of the packet header), the MAC can only be placed in the header after computing the MAC over the entire packet. That is, the computation of the MAC must be fully complete before the end of the packet header is transmitted. I illustrate this below:

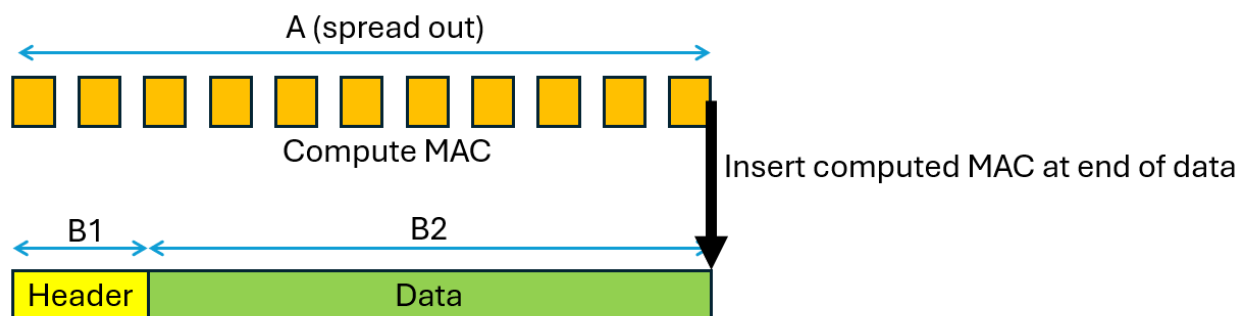


Houh Figure 1

As shown in each of the Houh Figures included herein, **A** is the time required to compute the MAC, **B1** is the time required to transmit the packet header, and **B2** is the time required to transmit the packet's data contents. As shown in Houh Figure 1, there is some parallel computation, but it occurs during the interval the header is being transmitted, during the time **B1** in the figure above. Note that during this time, the MAC is being calculated based on the end of the packet data. The transmission of the packet can begin before the MAC computation is completed, so that the computation is complete just in time to place the value into the packet header. In this

case, the packet transmission is complete ($A + B2$) seconds after the MAC computation starts.

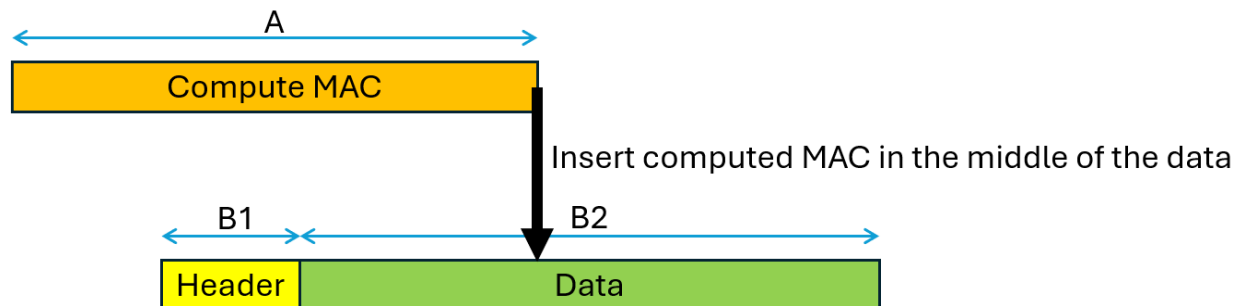
19. However, when the MAC is computed “on-the-fly” and placed at the end of a packet, the packet transmission can begin at the same time the MAC is computed. As shown in Houh Figure 2 below, the MAC is computed in parallel with the packet transmission and is completed just-in-time to be placed at the end of the packet. In this case, the packet is fully transmitted ($B1 + B2$) seconds after the MAC computation starts, and the transmission would be complete earlier than if the MAC were placed at the end of the packet header:



Houh Figure 2

20. My First Declaration includes a discussion of why a POSITA would not be motivated to place the MAC in the middle of the data and I refer the Board to that discussion. *See, e.g.,* EX1003, ¶178. Similar analysis applies to placing the MAC in the middle of the packet as to placing the MAC in the packet header. In either scenario, there is lower latency to finish transmission if the MAC is placed at

the end of the packet (measured from the time the MAC computation starts). This is illustrated in Houh Figure 3 below.



Houh Figure 3

21. Here in Houh Figure 3, since $A > B1$, and the insertion point requires the MAC computation to start before the header is transmitted, the packet is not fully transmitted until a time greater than $B1 + B2$ (the time the packet would be transmitted if the MAC were computed on-the-fly and placed at the end of the packet). Even if the MAC were placed deep enough into the middle of the packet data, such that computation did not need to start before the header began to be transmitted, requiring an additional field to indicate where in the middle of the data the MAC was inserted would lead a POSITA to favor placing the MAC at the end of the packet. Thus, a POSITA would have been motivated to place the MAC at the end of the packet.

C. Dr. Stubblebine's Understanding of OSI Layers is Overly Strict

22. Dr. Stubblebine incorrectly views the layers of the OSI stack as requiring strict boundary requirements between the layers, such as when passing data

from one layer to another. In doing so, Dr. Stubblebine states that a “POSITA would have understood that if a packet requires information to be computed (in Dr. Houh’s hypothetical, a signature), that information will always have to be computed before passing the PDU to the next layer.” EX2004, ¶108. I disagree. These strict boundary requirements between layers do not exist. The ’682 Patent itself states that,

[i]n practice, many applications do not always permit exact nomination of strict boundaries specifically between the upper protocol layers, particularly layers 5 to 7. By way of example, within the context of Internet telephony, i.e. use of the telephone service using the medium of the Internet, the three layers above the transport layer 4 combine to form an ‘application layer’ for which, in a similar way to in the implementation above, an ‘application protocol’ is used.

EX1001, 1:37-45.

This discussion of the merging of layers 5-7 belies Dr. Stubblebine’s strict requirement that “information will always have to be computed before passing the PDU to the next layer.” EX2004, ¶108.

23. Furthermore, other portions of the ’682 Patent specification also teach that Dr. Stubblebine’s strict boundaries between layers do not exist. The ’682 Patent states that “[t]he OSI reference model (OSI model for short) from the International Standards Organization (ISO) has seven layers, each of which has a different functionality (in terms of abstraction). In the OSI model, ... layer 4, a transport layer, uses a transport protocol.” EX1001, 1:22-33. The ’682 Patent continues that “within the context of Internet telephony, ... the three layers above the transport layer 4

combine to form an ‘application layer’ for which, in a similar way to in the implementation above, an ‘application protocol’ is used.” EX1001, 1:40-45. Yet, the ’682 Patent later states that “directly above the layer 4 is the real time transport protocol (RTP) 201” and teaches that application protocols are used above layer 4. EX1001, 6:39-40, FIG. 1. Under Dr. Stubblebine’s strict boundary requirements interpretation of the OSI layers, this presents many contradictions. For example, it is layer 4 that uses a transport protocol, but RTP, the Real-time Transport Protocol, is used above layer 4, where the ’682 Patent states that an application protocol is used.

24. Dr. Stubblebine also discusses layers working in conjunction with each other by stating that “[i]n the context of Ethernet, transmission refers to generating and transmitting electromagnetic waves on a physical medium at ISO layer 1 (in conjunction with the link layer ISO layer 2) where the PDU is a single bit.” EX2004, ¶128; EX1031, 167:3-10. If all functions at one layer had to be completed prior to passing data down to the next layer, the lower layer would not need to work in conjunction with the upper layer and instead could perform its own function independently. It is clear that Dr. Stubblebine’s strict boundary requirements between the layers are not supported by the ’682 Patent specification, nor by his own testimony.

25. Further contradicting Dr. Stubblebine's strict interpretation of boundary requirements between layers is that as of the priority date of the '682 Patent, a POSITA would have been familiar with hardware protocol accelerators, a basic concept in computer networking. For example, IETF RFC 1936, published in April, 1996, teaches that:

Efficient software implementations of the Internet Checksum algorithm are often embedded in data copying operations ([1], Section 2). This copy operation is increasingly being performed by dedicated direct memory access (DMA) hardware. As a result, DMA hardware designs are beginning to incorporate dedicated hardware to compute the Internet Checksum during the data transfer. This note presents the architecture of an efficient, pipelined Internet Checksum mechanism, suitable for inclusion in DMA hardware [8].

EX1032, 1.

26. Hardware protocol accelerators allow incomplete packets, such as those including uncomputed header checksum values, to be passed from upper layers down to network interface hardware that compute and fill in these values at a lower layer in the protocol stack. Basturk specifically teaches using this kind of accelerator hardware:

Computation of cryptographic functions can be greatly speeded up by use of dedicated hardware. Our design does not require such hardware to be available, but adopts various design principles that would allow this hardware, if available, to be exploited in the best possible way. On-the-fly processing is important in this regard as it facilitates the eventual off-loading to the hardware.

EX1006 at 7. Basturk applies this idea of using “dedicated hardware” to signature computations. Basturk teaches that “[t]he goal is to avoid slowing down the regular

flow of data, and allow the possible off-load of signature computations and verifications to dedicated hardware.” *Id.*, 12.

27. Dr. Stubblebine implies that design choice reasoning at one layer cannot be applied to other layers. *See* EX2004, ¶110. I disagree. For example, the frame check sequence (“FCS”) in the Ethernet protocol is a value that is appended to an Ethernet frame that is a calculation over the data in the frame itself (excluding the preamble, start frame delimiter, and the FCS itself). *See* EX1014, 26, 29. Received frames with a locally calculated FCS that do not match the value calculated at the transmitter are deemed invalid and discarded. *Id.* At the higher TCP layer, the header checksum value in the TCP header “provides a verification that the information used in processing internet datagram has been transmitted correctly” and that “[i]f the header checksum fails, the internet datagram is discarded at once by the entity which detects the error.” EX1013, 9. The same reasoning at one layer is applied at other layers as well. For example, the placement of the FCS at the end of the Ethernet frame enables the FCS value to be computed as the frame is transmitted and appended to the end of the frame. This concept can be applied to the check sequences or authentication sequences at another layer that are also placed at the end of that layer’s PDU. The MAC can be computed “on-the-fly” and placed at the end of an RTP packet while the FCS is computed on the fly and placed at the end of an Ethernet frame. Such trailers are nested just as headers are nested.

28. Furthermore, Dr. Stubblebine also states that “Basturk’s secure transport protocol is a transport level protocol that specifically operates at the transport layer, not at the application layer.” EX2004, ¶110. He incorrectly reasons that Basturk’s aim to add security functionality that is consistent with existing communication structure is accomplished by “expressly *not* implementing its security features at the application layer.” *Id.* (emphasis in original). But Dr. Stubblebine testified in his cross-examination that in Basturk, the data is signed at the application level. EX1031, 172:15-16 (“Q What level is the data signed at? A At the application level.”). These admissions contradict his earlier opinions in his declaration. For example, an application cannot sign the data without the session key. Basturk is clear that authentication occurs at the application layer. Basturk teaches that “[w]hen [application] *A* initializes *M_A* (by passing it [application] *B*’s identity and the shared key *a*) the machine initiates and executes an authentication and key exchange protocol with *M_B*.” EX1006, 9.

D. Basturk’s Secure Transport Protocol Is Not A Reliable Delivery Protocol

29. Dr. Stubblebine’s chain of reasoning for why Basturk’s protocol is a reliable transport protocol like TCP and would not be compatible with Handley’s internet conferencing architecture is based on unsupported conclusions. Dr. Stubblebine wrongly relies on Basturk’s use of the word “reliability” to reason that Basturk’s transport protocol “is a reliable delivery protocol like TCP.” (*See* EX2004,

¶¶ 111-112). Further, he states because Basturk’s transport protocol “is a reliable delivery protocol like TCP” that it “would therefore have similar end-to-end delay drawbacks.” (*Id.*; EX2004 at ¶ 112). And lastly, “that because Basturk’s protocol is a reliable delivery protocol like TCP, it would be unsuitable for transporting Handley’s real-time RTP traffic for the same reasons that Handley teaches TCP is unsuitable.” EX2004 at ¶114. I disagree with Dr. Stubblebine’s flawed chain of reasoning as directly contrary to the basic knowledge and understanding of a POSITA in this field.

30. First, Dr. Stubblebine bases his initial assumption on Basturk’s single use of the word “reliability” in its statement that “is essentially an extension of the standard transport level *reliability* feature.” EX2004, ¶111 (citing EX1006, 4). Then, Dr. Stubblebine ignores the written teachings of Basturk, keying in on the word “reliability” to analogize Basturk to other protocols that provide reliability, such as TCP. In full, Basturk states that “[s]ecure delivery, roughly speaking, is a guarantee to the sender A that his message was received by B and not intercepted and destroyed by the adversary. It is essentially an extension of the standard transport level reliability feature to the more adversarial setting we are discussing.” EX1006, 4. A POSITA would have understood Basturk is not teaching reliable transport features of TCP, but something totally different. Reliable delivery protocols like TCP provide the ability to “recover from data that is damaged, lost, duplicated, or delivered out

of order” as Dr. Stubblebine notes. EX2004, ¶ 111 (quoting EX2006, 10). In contrast, TCP does not guarantee to the sender A that his message was (1) “received by B,” (2) “not intercepted,” by the adversary, or (3) “destroyed by the adversary.” EX1006, 4. Dr. Stubblebine confirmed as much in his cross-examination. For example, when asked “[d]oes TCP guarantee protocol messages are not destroyed by an adversary?” Dr. Stubblebine answered “no . . . there is no guarantees.” EX1031, 197:6-9.

31. Basturk, titled “**Secure Transport Protocols** for High-Speed Networks,” is focused is on *secure* delivery, not reliable delivery. EX1006, 1. The reliability feature described in Basturk serves the purpose of authenticating the sender of the data, which is not a feature of protocols such as TCP. *See* EX1006, 4. This reliability feature of Basturk, e.g., to sign the packets (authenticating the source of data) is orthogonal to the reliability features in protocols such as TCP, which deliver the data reliably. *Id.*

32. By way of further example, Basturk describes that an adversary can forge a reply that acknowledges the receipt of data by the recipient and can intercept and destroy packets while sending these forged acknowledgements. *See* EX1006, 4. The sender believes the data was received when it was not. *Id.* Basturk teaches that this is possible because the acknowledgments are not generated by an application and, thus, are not signed. *Id.* Consequently, forged messages can be erroneously

accepted. *Id.* As seen in Basturk, forged data can be reliably delivered, but the delivery of forged data can be prevented by adding a signature to the data. In other words, verifying that the data came from the correct sender is orthogonal to the correct delivery of that data. Thus, the “reliability” features in Basturk (guaranteeing to the sender that its message was received, not intercepted, and not destroyed by and adversary) do not overlap with the reliability features of TCP (“recovering from data that is damaged, lost, duplicated, or delivered out of order”), which a POSITA would have known and understood. *Id.*; EX2006, 10.

33. As described in my First Declaration, and like Basturk’s forged messages example, Handley describes an example “in which authentication is needed in a multicast environment.” EX1003, ¶67. I provided a citation from Handley in my First Declaration that described a situation where a Rolling Stones concert was broadcast one of their live concert songs, at which time the band Severe Tire Damage started to broadcast one of their songs live to the same multicast group. *See* EX1003, ¶67; EX1005, 13. Handley states that “[c]learly commercial broadcasters want to avoid occurrences like this one.” EX1003, ¶67 (quoting EX1005,13). As noted in my First Declaration, “a POSITA would have understood and found obvious that Basturk’s authentication methods would have prevented unauthenticated interruption of video transmissions, like that in the Rolling Stones concert.” EX1003, ¶35. As described, the reliability feature in Basturk is to

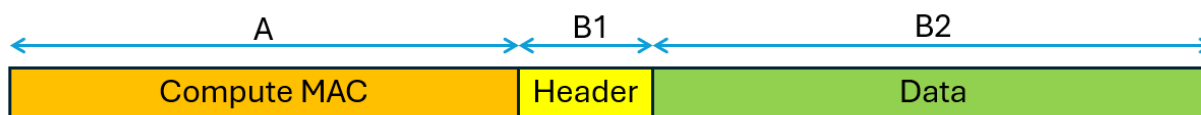
authenticate the sender of the data, which is not a feature of protocols such as TCP, whose role is to provide reliable delivery of data, i.e., deliver data accurately and in order over an unreliable network. Incorporating reliability features like those of TCP would not have prevented Severe Tire Damage from broadcasting to the same Rolling Stones multicast, though Basturk's authentication methods would have, which a POSITA would have known and understood.

E. Basturk's Secure Transport Protocol and Authentication Techniques are Compatible with Handley's Internet Conferencing Architecture

34. Basturk's secure transport protocol is compatible with Handley's internet conferencing architecture in part because Basturk's "on-the-fly" teaching does not add any latency to the data transmission. Dr. Stubblebine incorrectly equates Basturk's "reliability" feature to reliable protocol features like those found in TCP. *See* EX2004 at ¶113. Dr. Stubblebine's string of unsupported conclusions includes that Basturk's transport protocol "is a reliable delivery protocol like TCP" (EX2004, ¶¶111-112) and "TCP comes at the cost of end-to-end delay" (EX2004, ¶111). Dr. Stubblebine continues that since TCP introduces delays, that Basturk's transport protocol is incompatible with Handley because Handley teaches that "exceeding an end-to-end delay bound of a few hundred milliseconds renders the datagrams useless for the purpose they were intended" and "TCP traffic is normally not considered to be real-time traffic" (EX2004, ¶113 (quoting EX1005, 4-5)). I

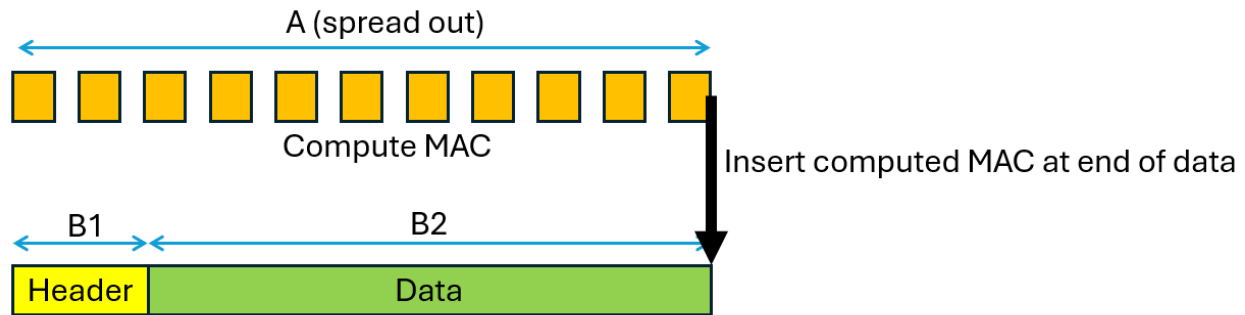
disagree. Basturk’s secure transport protocol is not a reliable delivery protocol like TCP and does not introduce hundreds of milliseconds of delay, as Dr. Stubblebine implies. EX2004, ¶113. As described in my First Declaration, performing the MAC computation “on-the-fly” as Basturk teaches does not add any latency to the transmission because the MAC computation occurs in parallel to the transmission. EX1003, ¶¶65, 85.

35. For comparison, I illustrate below a serial computation in Houh Figure 4 (no on-the-fly used) with a parallel computation in Houh Figure 5 (on-the-fly used). In Houh Figure 4, where on-the-fly is not implemented, the MAC is first computed, placed in the appropriate place in the packet, and then the packet is transmitted. For this serial computation (not on-the fly), **A** is the time required to compute the MAC, **B1** is the time required to transmit the packet header, and **B2** is the time required to transmit the packet’s data contents.



Houh Figure 4

Measured from the time the MAC computation is started, the packet is fully transmitted in (**A** + **B1** + **B2**) seconds. Compare this serialized computation in Houh Figure 3 to the parallel computation of the MAC as shown below in Houh Figure 5 (on-the-fly used):



Houh Figure 5

Basturk teaches “on-the-fly” computation as illustrated in Houh Figure 5. EX1006, 7-8, 12. In this example of “on-the-fly” computation, while the packet is being transmitted, the computation of the MAC begins and continues in parallel until the computation is finished. The computed MAC value is inserted at the end of the packet as transmission completes. Here, the time from when the MAC computation is started to when the packet transmission is finished is $(B1 + B2)$ seconds.¹ As seen,

¹ In this example, the assumption is that the system is limited by the transmission speed, rather than the compute cycles available. Even in the case this is not true, i.e., it takes longer to compute MAC than transmit the packet, it still will always be faster to transmit the packet using the on-the-fly approach than a serial approach. In this case the packet will be transmitted the lesser of (A) or $(B1 + B2)$ seconds after the computation of the MAC starts, which always occurs sooner than in the serial approach that Dr. Stubblebine incorrectly opines what Basturk teaches.

Basturk's "on-the-fly" MAC computation would not add any latency to Handley's internet conferencing architecture and would meet Handley's delay bounds. EX1005, 4-5.

36. This MAC computation, even if not performed in parallel, would not add hundreds of milliseconds of time to the latency as Dr. Stubblebine asserts. *See* EX2004, ¶113. Since Basturk teaches that MAC computation can be performed "on-the-fly," it would not take any longer than it takes to transmit the data than it would to compute the MAC. Note, that if the transmission of the packet itself already takes hundreds of milliseconds, the transmission itself already exceeds the delay bounds and adding MAC computation time does not make a difference between exceeding the delay bounds or not. And in the rare case where the MAC computation puts the transmission over the delay bounds, the transmission delay was already very close to the delay bounds, or the MAC computation takes an unreasonably long time. Neither of these situations is taught in Basturk since it teaches that dedicated hardware can be used to accelerate the computation of the MAC. EX1006, 7. Exceeding the delay bound of hundreds of milliseconds and equating that delay to the delay of MAC computation of Basturk (as Dr. Stubblebine suggests) implies a packet rate of about three to five packets per second, which is many orders of magnitude slower than actual packet transmission rates as of the priority date of the '682 Patent. For example, a 10-megabit Ethernet network can support transmission

of roughly 800 packets per second of the largest packets possible on an Ethernet—roughly 1,500 bytes (octets) or 12,000 bits. *See* EX1014, 58 (“maxFrameSize 1518 octets”). I also note that today, much of real-time video is transmitted using TCP, and the problems that Dr. Stubblebine highlights are not problems at all.

37. Patent Owner and Dr. Stubblebine misinterpret Basturk’s protocol, claiming it “similarly uses acknowledgements, and thus would be incompatible with Handley due to delays caused by acknowledgment and retransmission.” POR, 30 (citing EX2004, ¶116). But Dr. Stubblebine does not argue that use acknowledgements are incompatible with Handley due to delays. Instead, Dr. Stubblebine only states that “Basturk’s protocol similarly uses acknowledgements, and thus would be similarly disadvantageous and thus would not be combined with Handley for similar reasons.” EX2004, ¶116. Dr. Stubblebine’s opinion regarding acknowledgments is premised on Handley’s teaching that “exceeding an end-to-end delay bound of a few hundred milliseconds renders the datagrams useless for the purpose they were intended.” EX2004, ¶113 (citing EX1005, 4-5). But the main source of delay in receiving acknowledgements are due to the round-trip transmission delays between endpoints. Within the United States, e.g., between Boston and San Francisco, round-trip transmission delays could reach approximately 50 milliseconds. This is the time it takes for light to travel round-trip across the country, adjusted for the refractive index of optical fiber that slows the

speed of light. This ~50 milliseconds is far less time than the threshold contemplated by Handley of a “few hundred milliseconds.” EX1005, 4-5.

38. Furthermore, the combination I set forth is not based on the wholesale combination of both Handley and Basturk. I describe in my First Declaration that a “POSITA would have found it obvious to implement Basturk’s secure transport protocol and its authentication techniques to Handley’s internet conferencing architecture,” which includes Basturk’s teaching of computing the MAC “on-the-fly.” EX1003, ¶¶80, 84. Basturk teaches:

Integrity via signing. If the integrity option is selected then data is *signed* by M_A under the session key α (a signature is also called a MAC, or message authentication code). The signature covers not only the data payload but also the transport header (as explained later). On receiving a signed packet, M_B uses α to verify the signature.

EX1006, 9. A POSITA would not have understood this passage to mean that “[i]ntegrity via signing” only applies to acknowledgment packets, as Dr. Stubblebine opines. *Id.* Instead, a POSITA would have understood that Basturk’s plain teaching of “[i]ntegrity via signing” means any data sent is signed under the integrity option. This can include any data sent, even data that does not require acknowledgment. *Id.*

39. Further, Dr. Stubblebine incorrectly states that “Basturk teaches away from modifying the application layer with its transport level protocol.” EX2004, ¶117. In doing so, Dr. Stubblebine states “Basturk indicates that it would be ‘poor design,’ ‘belies the layered communication structure,’ would ‘adversely affect

performance,’ would ‘mak[e] implementation significantly harder,’ duplicate functionality’ and ‘degrade performance’” to “modify[] an application to implement its transport protocol.” *Id.*, ¶¶117-18. However, Dr. Stubblebine takes these statements out of context. Basturk’s footnote 3 states that “asking the application to duplicate these transport functions is usually poor design in that it belies the layered communication structure, and would also adversely affect performance.” EX1006, 4, n. 3. But the combination I set forth does not duplicate functions because Handley does not disclose the use of a MAC and any checksums in different protocol layers used in Handley are not for the purpose of authentication. Furthermore, I set forth the motivations a POSITA would have to make the combination above. *See* §§ IV.A-E, *supra*.

40. Furthermore, I discuss above how the ’682 Patent specification teaches that there are not such strict boundaries between layers. I incorporate my analysis from Section IV.C above. *See* § IV.C, *supra*.

F. Element [1.1]

41. Dr. Stubblebine opines that Handley or Basturk do not disclose Element [1.1] because “a POSITA would not consider encryption for the purposes of ‘confidentiality’ as implemented in RTP to be the same as authentication much less ‘providing transmitter-to-receiver authentication at a Real Time Transport Protocol (RTP) packet level as an application protocol on an application layer.’” EX2004,

¶119. I disagree. In my First Declaration, I state that a POSITA would have understood that “Handley discloses the implementation of authentication at the RTP packet level because using encryption schemes like DES or MD5 to manipulate the RTP packet directly occurs at the RTP packet level since Handley teaches that the encryption is specified by RTP to encrypt RTP packets and not at any other layer in the stack.” EX1003, ¶78. Because Handley discloses encryption at the application layer, for example, using an encryption scheme such as DES, Handley discloses authentication at the application layer. *See* EX1005, 12. DES is a symmetric key algorithm. This means the same key used to encrypt the data is used to decrypt the data. To successfully decrypt data using DES, the receiver must have the same symmetric key used to encrypt the data at the sender. If the data is successfully decrypted, then the receiver knows that data was sent by the sender, and thus the receiver has authenticated the sender.

42. I further stated that “[t]o the extent Handley does not expressly disclose Element [1.1], it is my opinion that this element would have been obvious in view of Handley in combination with Basturk.” EX1003, ¶79. I provided the reasoning that a POSITA would have been motivated to combine the teachings of Handley with the teachings of Basturk, and I discuss this combination and reasons to make this combination above in Sections IV.A-E, which I incorporate here. §§ IV.A-E, *supra*.

43. Dr. Stubblebine next states that “[Dr. Houh] identifies nothing in Basturk that discloses ‘providing transmitter-to- receiver authentication at a Real Time Transport Protocol (RTP) packet level as an application protocol on an application layer.’” EX2004, ¶120. I disagree. I do not rely on Basturk alone as rendering obvious or anticipating the claim, but rather I rely on the combination of Basturk and Handley. Dr. Stubblebine further states that “[a] POSITA would therefore not have found it obvious to provide authentication at an RTP packet level as an application protocol on an application layer in view of the combination of Handley and Basturk.” EX2004, ¶123, *see also* ¶¶121-122. I give reasons why Dr. Stubblebine is incorrect in Sections IV.C and IV.E above and incorporate that discussion here. *See* §§ IV.C, IV.E, *supra*.

44. In opining that a POSITA would not have found it obvious to combine the teachings of Handley and Basturk, Dr. Stubblebine states that “a POSITA would have understood that Basturk’s protocol, which Basturk makes clear is a transport level protocol below the application layer, does not provide authentication as an application protocol on an application layer.” EX2004, ¶125. I disagree. Dr. Stubblebine improperly looks at Basturk in isolation and not at the combination of Handley and Basturk. For example, in my First Declaration, I noted that authentication by “attaching to each packet a signature, computed under α , and taken over some portions of the header and the data payload,” as taught by Basturk, is

applied to the internet conferencing architecture of Handley. EX1003, ¶80. Thus, a POSITA would have applied the authentication at the RTP packet level, as a POSITA “would have understood that Handley discloses the implementation of authentication at the RTP packet level.” *Id.*, ¶78. Even though the specification of the ’682 Patent describes RTP as “the real time transport protocol” (EX1001, 6:39-40) it simultaneously is “directly above the layer 4” (*Id.*), which is above the transport layer 4 (*Id.* at 1:32-33). As previously described, the ’682 Patent also teaches that “[i]n practice, many applications do not always permit exact nomination of strict boundaries specifically between the upper protocol layers, particularly layers 5 to 7.” EX1001, 1:37-40. Thus, even though RTP is a “transport protocol,” the ’682 Patent teaches that the layers above the layer 4 transport layer are effectively application layer protocols. *Id.* As such, attaching a signature to the RTP packet in the combination of Handley and Basturk is “above the transport layer 4” and an application protocol as claimed by the ’682 Patent. Thus, a POSITA would have found it obvious to combine the teachings of Handley and Basturk.

G. Element [1.2]

45. Dr. Stubblebine states that “[a] person of ordinary skill in the art would have understood that this placement of context information is how the reduction in context retrieval latency for ‘on-the-fly’ processing would be achieved.” EX2004, ¶87, *see also* ¶127. However, Dr. Stubblebine is incorrect in his interpretation of

“on-the-fly” for the reasons discussed above in Section IV.A, which I incorporate here. *See* § IV.A, *supra*.

46. Dr. Stubblebine further states that “‘on-the-fly’ processing is enabled, in part, by locating the context information for signature verification (i.e., the session key α and cryptographic information) early in the packet, namely in the header, not by locating the signature (the value being computed with the context information) at the end of the packet in a trailer.” EX2004, ¶87. However, Dr. Stubblebine is incorrect for the reasons discussed above in Sections IV.A-E, which I incorporate here. *See* §§ IV.A-E, *supra*.

47. A POSITA would have been motivated to locate the MAC at the end of the packet. I discuss the reasons why above in Section IV.B, which I incorporate here. *See* § IV.B, *supra*.

48. Dr. Stubblebine states that “[a] person of ordinary skill in the art would have understood that the signature computation style depends on the algorithm selected, not where the signature value is located in the packet.” EX2004, ¶88. However, the context retrieval is unknown initially on the reception side. On the transmission side, the context is already known well in advance of the packet being transmitted. That said, not knowing the context upon packet reception still allows for computations using common hashing schemes before the context is retrieved, and locating the MAC at the end of the packet still allows for on-the-fly computation,

i.e., verifying the MAC in parallel as the packet is being received. If the context retrieved matches the hashing algorithm applied, the computation can continue in parallel to the packet reception and verify the MAC immediately as the MAC at the end of the packet is received. If not, the correct hashing method that matches the context can be restarted, and as the computation can be restarted and to the extent that the header data is a small percentage of the overall packet, the computation could complete as the MAC value at the end of the packet arrives. However, computing the MAC on reception also provides additional motivation to a POSITA to place the MAC at the end of a packet, in addition to the reason I previously stated, where the MAC is placed at the end of a packet. I discuss this above in Section IV.A and I incorporate that analysis here. *See* § IV.A, *supra*.

49. Dr. Stubblebine continues with “[t]he PDU needs to go through the process of encapsulation down the protocol stack until it reaches the physical layer, which is where bit-by-bit transmission on the physical layer occurs. ... This transmission at the physical layer, however, does not happen before the processing at higher layers is complete.” EX2004, ¶129. I disagree. Dr. Stubblebine is unreasonably strict regarding the boundaries between layers and provides no citation for support. The ’682 Patent is not as strict about boundaries between layers as Dr. Stubblebine opines, and there is no such requirement that would so teach a POSITA. I discuss this above in Section IV.C, and I incorporate that analysis here. *See* § IV.C,

supra. Dr. Stubblebine also states that “a POSITA would not have been motivated to add a trailer to an RTP packet or use on-the-fly processing based on the physical layer transmission process that Dr. Houh describes.” EX2004, ¶129. Dr. Stubblebine is incorrect and I discuss the reasons why in Section IV.C, which I incorporate here. *See* § IV.C, *supra*.

50. As a reason for why a POSITA would not have been motivated to add a trailer to an RTP packet, Dr. Stubblebine states that “RTP can look for information in the header to check for validity issues. This further demonstrates that a POSITA would not have been motivated to add a trailer to an RTP packet based on the location of error checking data in a lower level protocol.” EX2004, ¶134 (citing EX1008, 59). But the validity check to which Dr. Stubblebine cites is for the RTP header and not for the data contents of the RTP packet, as it is a “header validity check.” EX1008, 59, Section “A.1 RTP Data Header Validity Checks.” Dr. Stubblebine further states that “[t]o the contrary, RTP provides a method for extending the RTP header to support additional information, which can advantageously be ‘less expensive to process because it is not conditional nor in a variable location.’” EX2004, ¶134 (quoting EX1008, 14-15). Dr. Stubblebine implies that this teaches away from putting the MAC at the end of an RTP packet. But Dr. Stubblebine ignores that the RTP RFC 1889 specifically states “[n]ote that this header extension is intended only for limited use. Most potential uses of this

mechanism would be better done another way, using the methods described in the previous section.” EX1008 at 15. Thus, RFC 1889 directly contradicts Dr. Stubblebine’s opinion because the RTP header extension is conditional on the “X bit” in the RTP header having a value of one. EX1008, 15 (“If the X bit in the RTP header is one, a variable-length header extension is appended to the RTP header, following the CSRC list if present.”). Additionally, the RTP header extension is also a variable length field and thus is no faster to deal with than the overall data field being variable length (notwithstanding that the variable length of the RTP data is always variable length and always must be dealt with accordingly). *Id.*

51. Furthermore, even if there were already a place to add the MAC in the header, a POSITA would still have been motivated to place the MAC at the end of a packet and not in the header. I discuss the reasons above in Section IV.B and incorporate that analysis here. *See* § IV.B, *supra*.

H. Element [1.3]

52. Dr. Stubblebine states that “Dr. Houh concedes that Basturk does not disclose RTP, EX1003 ¶¶ 66, 85, 260, thus he has not shown that Basturk discloses ascertaining ‘whether the receiver knows the transmitter based on the RTP packet level authentication data,’ as the claim requires.” EX2004, ¶136. Dr. Stubblebine is incorrect as I do not rely on Basturk alone to render the challenged claims obvious.

Rather, I rely on the combination of Handley and Basturk, where Handley discloses RTP in the combination. As such, Dr. Stubblebine is incorrect.

53. Dr. Stubblebine does not offer any new analysis pertaining to Element [1.3] in EX2004. ¶137, and instead points to his analysis for Elements [1.1] and [1.2]. I have already discussed the reasons Dr. Stubblebine provides incorrect analysis for Elements [1.1] and [1.2] in Sections IV.F-G, and I incorporate that analysis here. *See* § IV.F-G, *supra*.

I. Element [1.4]

54. Dr. Stubblebine does not dispute that Handley and/or Basturk disclose and/or renders obvious Element [1.4]. As such, my opinions in my First Declaration apply and stand. *See* EX1003, ¶¶92-94.

J. Independent Claims 25-28

55. In stating “[Dr. Houh] has failed to show that claims 25-28 would have been obvious over the combination of Handley and Basturk,” Dr. Stubblebine incorporates his reasons with respect to Elements [1.1] – [1.4] but adds no additional analysis. I disagree with his opinions for the reasons explained above and opine that Handley and/or Basturk disclose and/or render obvious independent claims 25, 26, 27, and 28. *See* §§ IV.F-I, *supra*.

K. Dependent Claims 2-21, and 23.

56. Dr. Stubblebine provides no additional analysis for dependent claims 2-24 and instead states “. . . by their dependency on claim 1, claims 2-24 are

patentable over Handley in combination with Basturk.” EX2004, ¶139. I disagree with his opinions for the reasons explained above and opine that Handley and/or Basturk disclose and/or render obvious dependent claims 2-21, and 23. *See* §§ IV.F-I, *supra*.

V. GROUND II

A. Element [1.1]

57. Dr. Stubblebine states that I do not “show that PacketCable’s implementation of RTP uses the application level framing concept as described in the RTP specification” and that I do “not show that PacketCable’s alleged ‘adding a MAC’ (with or without RC4) is provided ‘as an application protocol on an application layer,’ as this element requires.” EX2004, ¶141. I disagree. As I stated above, the ’682 Patent itself teaches that RTP is above layer 4 and shows protocols above layer 4 as an “Application protocol” used by an “Application.” EX1001, 6:39-40, FIG. 1. I discuss this in more detail in Section IV.C above, which I incorporate here. *See* § IV.C, *supra*.

58. Dr. Stubblebine states that “PacketCable indicates to a POSITA that MTA devices generally are responsible for providing secure transport services, including ‘Authentication, confidentiality and integrity of some messages between the MTA and other PacketCable network elements.’” EX2004, ¶142. To the extent Dr. Stubblebine opines that a device performing authentication is different than an

application performing authentication, that is incorrect. The authentication may be for different uses, and the device performing authentication does not preclude applications on the device from performing authentication.

59. Also, as I stated in my First Declaration, “PacketCable explicitly discloses RTP, an application protocol whose packets are created at the application layer. EX1007, 45. For example, PacketCable discloses that ‘RTP packets are encrypted directly with RC4, without any additional security layers. An MMH-based MAC (Message Authentication Code) optionally provides message integrity.” EX1003, ¶174. Again, the ’682 Patent itself teaches that RTP is above layer 4 and shows protocols above layer 4 as an “Application protocol” used by an “Application.” EX1001, 6:39-40, FIG. 1. I discuss this in more detail in Section IV.C above, which I incorporate here. *See* § IV.C, *supra*.

B. Element [1.2]

60. Dr. Stubblebine admits that “PacketCable mentions that ‘[e]ach RTP packet may include an optional message authentication code (MAC)’ and that ‘MAC computation spans the packet’s unencrypted header and encrypted payload,’” but opines that “a POSITA would not have understood this to disclose (explicitly or inherently) inserting the optional MAC ‘at end of a whole RTP packet payload.’” EX2004, ¶147 (quoting EX1007, 47). I disagree. I discuss in my First Declaration that there are only a finite number of places where the MAC could be placed to start,

and why it would be obvious to a POSITA that the MAC would be placed at the end of a packet. *See* EX1003, ¶178. I also discuss this further in Section IV.B, which I incorporate here. *See* § IV.B, *supra*.

61. Dr. Stubblebine states that “Dr. Houh does not identify any negative impacts to ‘performance’ that would result from placing, for example, a MAC in ‘in the middle’ of a packet payload or even in the middle of a packet header.” EX2004, ¶150. Dr. Stubblebine is incorrect. I do discuss these reasons in my First Declaration. *See e.g.*, EX1003, ¶178. For example, I state that “the data in the packet would be interrupted by the authentication data, and extra data fields would be needed to identify the exact location of the authentication data within the packet data” as a negative impact to performance when the MAC is placed in the middle of a packet. EX1003, ¶178. Dr. Stubblebine misses or plainly fails to acknowledge that extra data in a packet means more storage is required. Extra data in a packet also means the transmission of the packet takes longer because for a fixed transmission rate, transmitting more data requires more time. I also stated that “placing authentication at the end of the payload would allow the transmitter to transmit data while simultaneously computing the checksum and preventing any unnecessary transmission delay by having to first compute the checksum and then transmit the payload.” *Id.* I further explain this above in Sections IV.B-C, which I incorporate here. *See* §§ IV.B-C, *supra*.

62. Dr. Stubblebine states that “Dr. Houh does not identify any ‘packet fragmentation issues’ that would favor putting data at the end of an RTP packet payload.” EX2004, ¶151. However, in my First Declaration, I state that by placing the MAC in the middle of the data, “the data in the packet would be interrupted by the authentication data, and extra data fields would be needed to identify the exact location of the authentication data within the packet data.” EX1003, ¶178. The authentication data in the middle of the packet data causes the data to be separated into two segments – the data segment before the authentication data, and the data segment after the authentication data. A POSITA would know and understand to avoid this fragmentation because extra data fields would be necessary to specify the location of the authentication data in between the two fragmented data segments. Furthermore, I state in my First Declaration that “[a] POSITA would have understood and found obvious that placing authentication at the end of the payload would allow the transmitter to transmit data while simultaneously computing the checksum and preventing any unnecessary transmission delay by having to first compute the checksum and then transmit the payload.” EX1003, ¶178. I further explain this above in Section IV.B, which I incorporate here. *See* § IV.B, *supra*.

63. When Dr. Stubblebine referred to “fragmentation” at his cross-examination, he appears to be limiting his view of “fragmentation” to the very specific issue of link fragmentation. *See, e.g.*, EX1031, 218:10-228:9. Link

fragmentation in a network occurs when a packet exceeds the Maximum Transmission Unit (“MTU”) of a link over which the packet is transmitted. For example, if the MTU of a particular link in the network is 512 bytes, and a packet is 1500 bytes, the packet must be fragmented in three units in order to be transmitted over a link. Thus, Dr. Stubblebine appears to be emphasizing and limiting his use of “fragmentation” to this scenario by focusing on the small sizes of packets. All to the exclusion of other forms of fragmentation.

64. Dr. Stubblebine continued to state that “[y]ou don't automatically get fragmentation if something is in the middle of a packet, but if it's a super short packet there is no fragmentation, RTP packets are potentially are very short.” EX1031, 24:22-25:4. Even considering Dr. Stubblebine's narrow understanding of “fragmentation” to apply only to the issue of link fragmentation, he is still incorrect about link fragmentation with respect to RTP packets. RTP packets have a 12 byte header (though it could be larger if header extensions are used), and are carried in a UDP/IP packet. A UDP header is 8 bytes long and an IP header is 20 bytes long (though it could be longer). Just the packet headers in an RTP packet are 40 bytes long at least. Dr. Stubblebine also proposes using the RTP header to store the MAC, which adds at least 8 additional bytes (32 bits for profile/length and minimum 32 more bits minimum to hold the MAC) to the RTP header. *See* EX1008, 14-15. Dr. Stubblebine's packet header is thus 48 bytes long before counting any data carried

by the packet. As of the priority date of the '682 Patent, ATM links were common in enterprise networks and the Internet. For example, U.S. Patent No. 6,526,056, filed on December 21, 1998, teaches "Wide-area enterprise networks are now more likely to be built on top of frame-relay and ATM networks than on top of circuit-switched (telephone) networks. Whereas a telephone network really provides circuits between backbone routers, a frame-relay or ATM network provides 'virtual circuits' between backbone routers." EX1034, 1:66-2:4. ATM cells (the name for a PDU in ATM) carry only 48 bytes (an ATM link's MTU), and thus any packet over 48 bytes carried over an ATM link is guaranteed to be fragmented into multiple ATM cells. Given that Dr. Stubblebine's modified RTP packets (that include the MAC in the header using the RTP header extension) are at least 48 bytes long, they are guaranteed to be fragmented when traversing any ATM link.

65. Furthermore, when Dr. Stubblebine was asked if he was familiar with the term "disk fragmentation" he said that he understood it to mean that the data in a file is stored in non-contiguous sectors on a disk: "Q. Have you ever heard of disk fragmentation? A Which? Q Disk fragmentation? ... A Like hard disk? Q. Sure. A. Yes. Q. And what is that? A. That's not something I opined on. Q. But you're familiar with it? A. I have some familiarity with it. Q. What is that general familiarity? A. There is read/write sectors on a disk, and sometimes content is distributed on different parts of the -- different sectors on the medium. And to read a file, you may

have to search on multiple sectors to retrieve the entire file. Multiple noncontiguous sectors.” EX1031, 220:80-229:6. Dr. Stubblebine agrees that fragmented can mean non-contiguous in the context of a disk drive. That is the sense of fragmentation as I discuss in my declaration, applied to data within a packet. When the MAC is put in the middle of a data section of the packet, it separates the data (that was originally contiguous in the packet) into two non-contiguous sections within a packet. Dr. Stubblebine appears to have an understanding of the word “fragmentation,” but he chooses to re-interpret my declaration’s use of “fragmentation” into his own narrow “link fragmentation” sense.

66. Dr. Stubblebine further opines that “[t]he point is, is that the transmitter has complete control over the size of the packet and -- how the PDU and the resulting IP packet. And so the transmitter knows how to structure the PDU so there is no fragmentation.” EX1031, 225:19-226:1. However, the transmitter does not know the makeup of the network between it and every other possible destination when traversing the Internet. Paths through the network over which the IP packet would travel would not be known, nor can be known the type of physical links present in the network, or the virtual paths traveled through network tunnels or their underlying physical equipment makeup. And, as discussed above, any of these paths or tunnels may include traversing an ATM link, over which Dr. Stubblebine’s modified RTP

packets are guaranteed to be fragmented into multiple ATM cells and thus, there is no way to avoid Dr. Stubblebine's fragmentation.

67. Dr. Stubblebine further states that "RTP does not define a trailer and instead defines a mechanism for placing additional information in the header, as explained above." EX2004, ¶154. He implies that this teaches away from putting the MAC at the end of an RTP packet. But Dr. Stubblebine ignores that the RTP RFC specifically states "[n]ote that this header extension is intended only for limited use. Most potential uses of this mechanism would be better done another way, using the methods described in the previous section." EX1008, 15. Dr. Stubblebine does not consider better ways of storing the MAC other than in the RTP header RTP header extension even though the RTP RFC explicitly urges the reader to do so. EX1008, 14 ("Additional information that is required for a particular payload format, such as a video encoding, should be carried in the payload section of the packet.").

68. Furthermore, there can only be one header extension in an RTP packet header, so this technique cannot be used where an RTP header extension already exists. EX1008, 15 ("Only a single extension may be appended to the RTP data header."). Also, the RTP header extension is both conditional and variable length when present. *Id.*, 15 ("If the X bit in the RTP header is one, a variable-length header extension is appended to the RTP header, following the CSRC list if present."). Dr. Stubblebine opines that the MAC would be placed in the header: "Like I keep saying,

putting it in -- even in this case, but separate from the case, putting it in a fixed location in the header is -- or is the easiest way to determine the location of the MAC and the processing to location of a MAC without having to deal with variable size PDUs.” EX1031, 234:16-22. But this is contradictory. Put plainly, Dr. Stubblebine proposes that using the RTP header extension for the MAC avoids dealing with variable sized PDUs by proposing using the RTP header extension, which itself is variably sized and part of the PDU. In fact, Dr. Stubblebine’s proposal to locate the MAC in the RTP header extension results in additional processing of variable-length data, not less. This is because using the RTP header extension introduces an additional variable-length field which was not used prior to being used for the MAC. Furthermore, packets are already variably sized – both the IP and UDP packet headers contain “length” fields. That means the variable length payload must already be dealt with and adding the MAC at the end would not result in dealing with variable size PDUs any more than is already done. But, adding an additional variable length field in the RTP does add a new variable-length field, i.e., the RTP header extension field adds more processing compared to only dealing with the variable-length IP and UDP data fields.

69. And, in another contradiction, Dr. Stubblebine states that “a POSITA would have understood that there are to be headers comprised of fixed length or length-determinal portions including the extended portion make it computationally

easier and more efficient to find the data as opposed to the payload area, which is variable and undefined length and require more computational resources to find the data there.’ So this is an example where I considered both the header, and I made comments about where if it wasn’t in the header.” EX1031, 219:3-13. But placing the MAC in the RTP header using the RTP header extension uses a variable length field, causing the RTP header to become a different length than normal, so it is not fixed length. And as I discuss above, IP and UDP packets are already variable length, so putting the MAC at the end does not use any more resources. This is because the end of the packet already has to be determined to know where the data ends, and knowing that the data ends with the MAC uses fewer computational resources than adding a variable-length RTP header extension at the end of the RTP header but before the data.

70. Furthermore, even if there were already a place to add the MAC in the header, a POSITA would still have been motivated to place the MAC at the end of a packet and not in the header. I discuss the reasons above in Section IV.B and incorporate that analysis here. *See* § IV.B, *supra*.

71. Dr. Stubblebine states that “RTP can rely on the underlying UDP protocol for providing checksum services, EX1008 at 3; § X.A.2.b, *supra*, thus a POSITA would not have been motivated to include a trailer at the end of an RTP packet based on the use of checksums in lower layer protocols.” EX2004, ¶155. I

disagree. The checksum field in a UDP header is a checksum of the data. While it can be used to verify that there are no errors in the received data, it is not used for authentication. This issue has no bearing on PacketCable's explicit statement that a MAC can be used in conjunction with RTP packets (EX1007, 45) and that it would have been obvious to put at the end of a packet. I discuss the reasons above in Section IV.B and incorporate that analysis here. *See* § IV.B, *supra*.

72. Dr. Stubblebine provides numerous opinions against putting the MAC at the end of a packet. *See* EX2004, ¶156. I disagree with each of them. I addressed these in my First Declaration (EX1003, ¶¶176-179) and above in Sections IV.B and V.A-B which I incorporate here. *See* §§ IV.B, V.A-B, *supra*.

73. Dr. Stubblebine then states:

To the extent Dr. Houh is suggesting that “placement of authentication data at the end of the whole RTP packet payload” somehow results in bit-by-bit processing benefits occurring across all layers going down to the physical layer where transmission occurs bit-by-bit; he hasn't shown this. This is contrary to how network protocol stacks operate. PDUs are created at higher layers and passed to lower layers for encapsulation.

EX2004, ¶160. Dr. Stubblebine misinterprets my First Declaration. I describe the benefits of putting the MAC at the end of a packet, but I do not describe it as “bit-by-bit processing benefits occurring across all layers going down to the physical layer” as Dr. Stubblebine states. In my First Declaration, I describe performance benefits that include reducing the latency to transmit packets. *See* EX1003, ¶178. I

also further explain the reasons above in Sections IV.B-C and incorporate that analysis here. *See* § IV.B-C, *supra*.

74. Dr. Stubblebine states that “[t]o the extent Dr. Hough [sic] is suggesting that somehow bit-by-bit processing occurs at one or more layers above the physical layer; he hasn’t shown how this decreases latency in transmission at the physical layer (where transmission occurs).” EX2004, ¶161. I describe the benefits of putting the MAC at the end of a packet in my First Declaration, but I do not describe it as “bit-by-bit processing” as Dr. Stubblebine infers. *Id.* In my First Declaration, I describe performance benefits that include reducing the latency to transmit packets. *See* EX1003, ¶178. I also further explain the reasons above in Section IV.A-C and incorporate that analysis here. *See* § IV.A-C, *supra*.

75. Dr. Stubblebine further states that “it makes no sense to pass data bit by bit from the transport layer (e.g. UDP or TCP) to the networking layer (e.g. IP). (Furthermore, it makes no sense to create IP, UDP, or TCP payloads of individual bits.) This is not how protocol stacks are implemented.” EX2004, ¶161. It is not clear why Dr. Stubblebine makes some artificial distinction between passing data bit-by-bit and passing data by any other method. Data in a computer is stored as bits and bits are grouped into bytes. Bytes are further grouped into words and long words. Passing any data requires the passing of bits (or pointers to those bits) and the task

of managing system state information. Regardless, I have addressed these issues above in Section IV.C and I incorporate that analysis here. *See* § IV.C, *supra*.

C. Element [1.3]

76. Dr. Stubblebine provides no additional analysis for Element [1.3] other than “[f]or the reasons I explained above, Dr. Houh has not shown that it would have been obvious to a POSITA to combine PacketCable and Handley to arrive at the requirements of Element [*sic*] and [1.2].” EX2004, ¶165. I disagree for the reasons explained above, which I incorporate here. *See* §§ V.A-B, *supra*.

D. Element [1.4]

77. Dr. Stubblebine does not dispute that PacketCable and/or Handley disclose and/or renders obvious Element [1.4]. As such, my opinions in my First Declaration apply and stand. *See* EX1003, ¶¶184-186.

E. Independent Claims 25-28

78. Dr. Stubblebine provides no additional analysis for Independent Claim 25-28 other than “Petitioners have failed to show that claims 25-28 would have been obvious over the combination of PacketCable and Handley for the reasons explained above with respect to Elements [1.1]-[1.3].” EX2004, ¶166. I disagree for the reasons explained above, which I incorporate here. *See* §§ V.A-D, *supra*.

F. Dependent Claims 22 and 24

79. Dr. Stubblebine states that “Dr. Houh does not explain whether or how the act of switching data ‘between a particular IP flow and a particular PSTN port

on the gateway’ involves ‘inserting... authentication data at end of a whole RTP packet payload’” EX2004, ¶167. I disagree. I explain in my First Declaration that a gateway “terminates the IP data side into the PSTN and ultimately acts as a switch between a particular IP flow and a particular PSTN port.” EX1003, ¶224. Since the gateway terminates the IP flow, it terminates the RTP streams, and this both generates RTP packets containing data from the PSTN side of the call and receives RTP packets to convert to the PSTN format. Thus, the gateway both (1) generates RTP packets with the MAC at the end of the packet when converting call data from the PSTN into IP packets, and (2) authenticates received RTP packets by computing the MAC and comparing it to the MAC received at the end of a packet when converting call data from IP packets to the PSTN.

G. Dependent Claims 2-21 and 23

80. Dr. Stubblebine provides no additional analysis for dependent claims 2-24 and instead states “[] by their dependency on claim 1, claims 2-24 are patentable over PacketCable in combination with Basturk.” EX2004, ¶182. I disagree with his opinions for the reasons explained above and opine that PacketCable and/or Handley disclose and/or render obvious dependent claims 2-24. *See* §§ V.A-D, *supra*.

VI. GROUND III

A. Motivation to Combine

81. Dr. Stubblebine’s arguments regarding the combination of PacketCable and Basturk are the same as his arguments against the combination of Handley and

Basturk. PacketCable also discloses RTP and further explicitly discloses using a MAC in conjunction with an RTP packet. In my First Declaration, I state that “PacketCable and Basturk are compatible or related because both implement MACs for authentication. EX1007, 45, 47; EX1006, 9, 11-13.” EX1003, ¶261.

82. The reasons why I disagree with Dr. Stubblebine are stated above with respect to Handley and Basturk in Section IV, and to the extent they apply in the same way to the combination of PacketCable and Basturk, I incorporate them here. *See* § IV, *supra*.

B. Independent Claim 1

83. Furthermore, Dr. Stubblebine reasons a MAC based on MMH is incompatible with Basturk because “Basturk requires the ability to start processing the message before the cryptographic key needs to be used because the claimed benefit is to reduce latency by performing some computation while waiting for context retrieval.” EX2004, 175. As noted above in Section IV.A, Basturk’s “on-the-fly” minimizes latency by computing the MAC in parallel to data transmission. *See* §IV.A, *supra*. Dr. Stubblebine then incorrectly states “a context retrieval would be required before processing any portion of the message using the secure MAC based on MMH. Consequently, the delay associated with context retrieval would be unavoidable in PacketCable, meaning that ‘on-the-fly’ cannot be achieved.” EX2004, 177. But Dr. Stubblebine’s testimony undermines his reasoning here.

During his cross-examination, when asked “Do you disagree with Basturk when it says, ‘It must be typically retrieved upon receipt of a packet, and this context retrieval step can introduce latency’?” Dr. Stubblebine answered:

I think the paper is inaccurate when it comes to it describing latency benefits. And it essentially admits it right here when it says, “Immediately availability when transmit.” So now, elsewhere in the paper, there is no issue of -- of -- elsewhere in the paper, there is supposedly an issue of latency for context retrieval. But now there isn't any for transmission. And for the same reasons there isn't any latency for transmission, there is also no latency for reception. And the reason being is, if you look -- at least one reason -- if you look at the context of -- of Basturk, it's using authenticated key exchange protocol. And this is a protocol where parties communicate to establish a key. And when that occurs, both a sender and receiver have a key in memory. And when the next packet happens, there is no context retrieval. There is -- the key is immediately available. Even if you want to call that “context retrieval,” it's nothing. It's insignificant. It's already in memory.

EX1031, 148:9-149:13; *see also Id.*, 146:11-147:7. Dr. Stubblebine admits that context is immediately available on both transmission and reception, which directly contradicts his statements that “the delay associated with context retrieval would be unavoidable in PacketCable, meaning that ‘on-the-fly’ cannot be achieved.” EX2004, ¶177.

C. Independent Claims 25-28

84. In stating “[Dr. Houh] has failed to show that claims 25-28 would have been obvious over the combination of PacketCable and Basturk,” Dr. Stubblebine incorporates his reasons with respect to Elements [1.1] – [1.3] but adds no additional

analysis. I disagree with his opinions for the reasons explained above and opine that Handley and/or Basturk disclose and/or render obvious independent claims 25, 26, 27, and 28. *See* §§ VI.A-B, *supra*.

D. Dependent Claims 22 and 24

85. Dr. Stubblebine provides no additional analysis for dependent claims 22 and 24 other than “[f]or the same reasons that I explained for claim 22 of Grounds II, Dr. Houh has not shown that the combination of PacketCable, and Basturk discloses or renders obvious claim [22 and 24].” EX2004, ¶¶180-181. I disagree for the reasons explained above and opine that PacketCable and/or Basturk disclose and/or render obvious dependent claims 22 and 24. *See* § V.F, *supra*.

E. Dependent Claims 2-21 and 23

86. Dr. Stubblebine provides no additional analysis for dependent claims 2-24 and instead states “[] by their dependency on claim 1, claims 2-24 are patentable over Handley in combination with Basturk.” EX2004, ¶139. I disagree with his opinions for the reasons explained above and opine that PacketCable and/or Basturk disclose and/or render obvious dependent claims 2-24. *See* §§ VI.A-B, *supra*.

VII. GROUND IV

87. Dr. Stubblebine provides no additional analysis on this Ground other than “[f]or the same reasons that I explained for claim 22 of Grounds II and III, Dr. Houh has not shown that the combination of PacketCable, Handley, and Basturk discloses or renders obvious claim [22 and 24].” EX2004, ¶¶183-184. I disagree. As

I explained in my First Declaration, “a POSITA would have been motivated to add the media gateway of PacketCable to the combination of Handley and Basturk.” EX1003, ¶320. For the reasons described above in Section V.F, and in my First Declaration, claims 22 and 24 are rendered obvious over Handley in combination with Basturk and PacketCable. *See* § V.F, *supra*; *see also* EX1003, ¶¶320-323.

Date: June 15, 2024

By: Henry H. Houh.
Henry H. Houh