

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

PALO ALTO NETWORKS, INC.,
Petitioner

v.

BT AMERICAS INC.,
Patent Owner

Case No. IPR2023-00889
U.S. Patent No. 7,895,641

**DECLARATION OF KEVIN JEFFAY, PH.D., IN SUPPORT OF
PETITIONER'S REPLY**

Mail Stop PATENT BOARD
Patent Trial and Appeal Board
U.S. Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

PAN EX1040
Palo Alto Networks v. BT Americas
IPR2023-00889

TABLE OF CONTENTS

I. Introduction 1

II. Legal Standards 3

III. Response to Certain Arguments Raised by Patent Owner and Dr. Lee 3

 A. Dr. Lee misinterprets Duvall’s disclosure associated with Figures
 3 and 4. 3

 B. A POSA would have understood how to implement an “analyst
 system,” as recited in independent claim 1..... 8

 C. Commercial database systems allowed for dynamic modification..... 10

 1. At the time of the ’641 patent, commercial database systems
 implemented multiversion concurrency control schemes. 10

 2. Exemplary commercial database-management system –
 Oracle 8i 13

IV. Conclusion.....16

I. INTRODUCTION

I, Kevin Jeffay, Ph.D., declare as follows:

1. I am over the age of twenty-one (21) years, of sound mind, and capable of making the statements set forth in this Declaration. I am competent to testify about the matters set forth herein. All facts and statements contained herein within my personal knowledge are true and correct, and I believe those items stated herein on information and belief to be true and correct.

2. I submitted a prior declaration in this matter (EX1003), which I understand was filed on April 28, 2023. I have been retained on behalf of Palo Alto Networks, Inc. (“Petitioner”) for the above-captioned *inter partes* review proceeding to provide my expert opinions and expert knowledge. I understand that this proceeding involves U.S. Patent No. 7,895,641 (“the ’641 patent,” EX1001), which is titled “Method and system for dynamic network intrusion monitoring, detection and response,” and is currently assigned to BT Americas Inc.

3. My background and qualifications were provided in paragraphs 11-22 of my prior declaration, and my CV was provided as EX1032. My statements in paragraphs 6 and 18 of my prior Declaration regarding my review of the ’641 patent and related materials remain unchanged, as do my understandings of the relevant legal principles stated in paragraphs 24-42.

4. Since my prior declaration, I have reviewed and considered the following additional materials:

<i>Paper #</i>	<i>Description</i>
10	Decision Granting Institution (“DI”)
15	Patent Owner’s Response (“POR”)

<i>Exhibit #</i>	<i>Description</i>
1041	Bernstein, P.A. and Goodman, N., “Concurrency Control in Distributed Database Systems,” <i>Computing Surveys</i> , 13(2):185-221 (June 1981) (“Bernstein81”)
1042	Bernstein, P.A. and Goodman, N., “Multiversion Concurrency Control—Theory and Algorithms,” <i>ACM Transactions on Database Systems</i> , 8(4):465-483 (December 1983) (“Bernstein83”)
1043	Reed, David P., “Naming and Synchronization in a Decentralized Computer System,” Thesis—Massachusetts Institute of Technology (1978)
1044	“Oracle 8i Concepts – Release 2 (8.1.6)” (December 1999)
1045	“List of databases using MVCC,” Wikipedia, accessed at https://en.wikipedia.org/wiki/List_of_databases_using_MVCC
1046	Deposition Transcript of Dr. Wenke Lee (April 4, 2024)
1047	“IP Datagram Encapsulation,” The TCP/IP Guide, available at http://www.tcpipguide.com/free/t_IPDatagramEncapsulation.htm
1048	IPR2019-01324, EX2001 (Lee Declaration)
1049	NetRanger Guide V. 1.3.1
1050	“Sun, Cisco & Pilot Form Marketing/Development Alliance” (1999)
1051	Ullman, J.D., “Principals of Database Systems,” <i>Computer Science Press, Inc.</i> , 2nd Ed. (1982)
2007	Deposition Transcript of Dr. Jeffay
2016	Declaration of Wenke Lee in support of Patent Owner’s Response

5. I have also considered all other materials cited herein. To the best of my knowledge, any new materials provided in support of this declaration are true and accurate copies of what they purport to be, and an expert in the field would reasonably rely on them to formulate opinions such as those set forth in this declaration.

6. I am being compensated at my customary rate of \$750 per hour for my work on this case. My compensation is not dependent upon my opinions, my testimony, or the outcome of this case.

II. LEGAL STANDARDS

7. I have also relied upon various legal principles (as explained to me by Petitioner's counsel) in formulating my opinions. My understanding of these principles is summarized in my prior declaration in paragraphs 24-42.

III. RESPONSE TO CERTAIN ARGUMENTS RAISED BY PATENT OWNER AND DR. LEE

A. Dr. Lee misinterprets Duvall's disclosure associated with Figures 3 and 4.

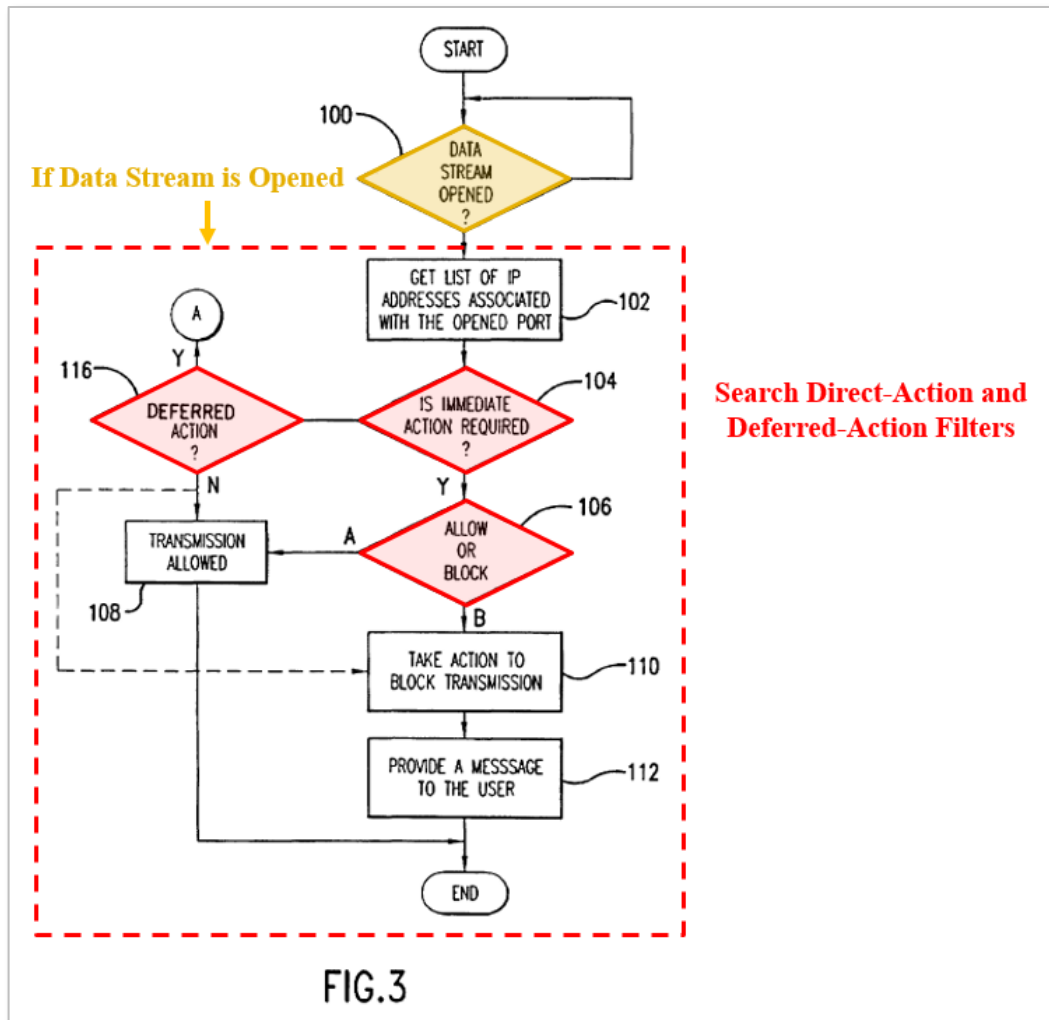
8. Dr. Lee asserts: "Petitioner's expert appears to be reading certain portions of Duvall's disclosures, completely out of context, imagining an embodiment that operates completely differently than how Duvall describes itself." EX2016, ¶59. I disagree. My opinions follow Duvall's express disclosure.

9. Dr. Lee acknowledges that "Duvall's direct-action filters are applied according to Figure 3," and "Duvall's deferred-action figures are applied according

to Figure 4.” EX2016, ¶64. Dr. Lee, however, incorrectly asserts that “Figures 3 and 4 are actually processing different transmissions in a sequence of transmissions.”

EX2016, ¶64. In my opinion, Dr. Lee misreads Duvall. Figures 3 and 4 of Duvall are part of the *same* process, which only executes *after* determining that a data stream is opened. EX1004, FIGs. 3, 4.

10. Specifically, Figure 4 of Duvall is clearly indicated as being a continuation of Figure 3, which executes after a determination is made in Figure 3 that a deferred action is to be applied. In addition, Duvall’s Figure 3 begins at step 100 where “[t]he filtering system detects *when the client opens the data stream* for a particular port and IP address,” as shown in my annotated Figure 3 below. EX1004, 4:40-42 (emphasis added).



EX1004, FIG. 3 (annotated).

11. Only after the process determines that a stream is opened does the remainder of Figure 3 execute. Then, once a stream is opened, for a message transmitted through the data stream, “retrieved filters are checked to determine if any require immediate action” at step 104, and “[i]f, at step 104, it is determined that no immediate action must be taken, it is determined whether a deferred action must be taken with respect to any of the retrieved filter[s] (step 116).” EX1004, 4:48-50, 4:65-5:1 (emphasis added). In other words, the figures expressly show that

both direct- and deferred-action filters are searched for the *same* message transmitted through the data stream, shown by steps 104, 106, and 116, highlighted above. And this operation is consistent with other portions of Duvall's disclosure, which explains that direct- and deferred-action filters are both checked with respect to the same message:

When a message is transmitted, whether that message is incoming or outgoing with respect to the client computer, the filtering system compares the IP address and/or other information in the data stream to the filter entries stored in the database to determine whether some action needs to be taken. The filters are preferably stored so that the system searches ALLOW filters first, BLOCK filters next, and deferred action filters last, and takes action based on the first matched filter.

EX1004, 4:23-37 (emphasis added).

12. Dr. Lee also incorrectly states that “Duvall’s direct-action filters are applied when the system *opens* a TCP stream, rather than when transmissions are sent or received *through* an already opened TCP stream.” EX2016, ¶78 (emphasis in original). Dr. Lee’s statement ignores the scenario illustrated in Figure 3 in which a data stream *is* opened and a subsequent message is transmitted through the stream (i.e., Duvall’s proceeds past step 100 of Figure 3). In that scenario, Duvall expressly discloses, “[w]hen a message is transmitted, . . . the filtering system compares the IP address and/or other *information in the data stream* to the filter entries stored in the database,” including *both* direct- and deferred-action filters. EX1004, 4:23-37

(emphasis added); *see also* EX1003, ¶¶129-36. Although Duvall discloses one way to block transmissions by “not executing the open command,” EX1004, 4:56-58, Duvall still discloses that its process depicted in Figures 3 and 4 evaluates messages transmitted through opened data streams against both direct- and deferred-action filters. EX1004, 4:23-37, 4:48-50, 4:65-5:1.

13. Finally, Dr. Lee asserts, “it appears Petitioner’s expert is relying on a system that effectively performs deep packet inspection to filter *packets* rather than the application-level transmissions actually disclosed by Duvall.” EX2016, ¶96 (emphasis in original). Dr. Lee ignores the opinions set forth in my original declaration, as well as the language of the ’641 patent claims.

14. First, my original declaration does not mention “deep packet inspection.” EX2016, ¶96. Instead, I opine that Duvall’s “filtering system compares portions of incoming and/or outgoing *messages* with filtering information.” EX1003, ¶126 (citing EX1004, 1:30-35) (emphasis added); *see also* EX1003, ¶¶127-128 (“As described for Figure 3, which I have reproduced below, Duvall uses ‘filtering’ to determine whether to allow or block *data transmissions . . .*”) (citing EX1004, 3:64-66, 4:37-42) (emphasis added). I further explained in my original declaration how messages are transmitted in Duvall. EX1003, ¶122. Specifically, Duvall discloses that “[m]essages are transmitted over the Internet among computers with a transport protocol known as Transmission Control Protocol/Internet Protocol

(TCP/IP).” EX1004, 2:42-44; EX1003, ¶122. A POSA would have understood that when using TCP/IP, as in Duvall, information is transmitted as IP datagrams, which are often referred to simply as “packets.” EX1047, 1-2. And those packets include the “application-level” data (e.g., message content) that Dr. Lee attempts to distinguish. EX1047, 1-2.

15. Second, the ’641 patent claims simply recite analyzing “status data” and “post-filtering residue,” regardless of how that information is derived from a data transmission. EX1001, Claim 1. For this reason, Duvall teaches the claim language regardless of whether it analyzes application layer messages as a whole or individual packets because Duvall teaches analyzing the claimed “status data” and “post-filtering residue.”

B. A POSA would have understood how to implement an “analyst system,” as recited in independent claim 1.

16. Dr. Lee takes issue with the evidence provided with my prior declaration that discusses existing third-party security-monitoring services. EX2016, ¶128 (discussing EX1037). As I explained in my prior declaration, IBM provided an outsourced network-security service coupled to its NetRanger product that included a “network operations center” and its own team of “security experts.” EX1003, ¶¶113, 142; EX1037, 277. NetRanger is just one example, and other similar services were present in the market before the ’641 patent. *See, e.g.*, EX1050 (discussing “Pilot Network Services” in 1999: “security experts monitor the sophisticated

technology 24×7 and continually develop upgrades to detect and repel attacks aimed at Pilot customers”).

17. Dr. Lee argues that “EX1037 provides no actual disclosure of *how* such outsourcing was accomplished, nor any disclosure of the systems used by the third-party.” EX2016, ¶128. But a POSA would have understood how to use third-party security-monitoring services, including, for example, how to transmit data to the third-party service for analyst review and response. The 1997 NetRanger User’s Guide, for example, provides a detailed communications architecture in which a “Director provides monitoring and analysis services to NetRanger, and communicates . . . via the communication system.” EX1049, 021; *see also id.*, 020 (“The NSX is the sensing and management component of the NetRanger System that resides on a corporate network. It communicates with one or more remote Director systems via the Post Office network communications system.”).

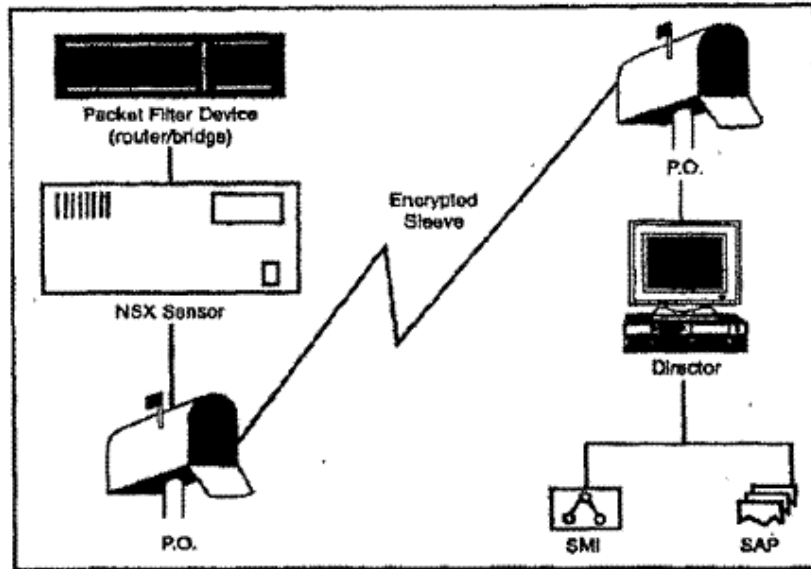


Figure 1.1: Basic NetRanger Components

EX1049, 020 (FIG. 1.1).

18. In this example, the Director system “provide[s] centralized command and control of an organization’s security perimeter” through “monitoring, management, data collection, data analysis, and user-defined actions.” EX1049, 032. And, like Duvall, NetRanger’s components “plug into TCP/IP networks.” EX1049, 019. Therefore, in my opinion, a POSA would have been well aware of how to accomplish outsourcing of information concerning Duvall-Chu’s data transmissions to trained network analysts, e.g., using standard network transmission techniques.

C. Commercial database systems allowed for dynamic modification.

1. At the time of the ’641 patent, commercial database systems implemented multiversion concurrency control schemes.

19. Dr. Lee asserts, “there is [] no evidence that a POSA would have known how to modify Duvall’s system at the time of invention to perform dynamic

modification.” EX2016, ¶141. But Dr. Lee ignores that Duvall’s system relies on “known database editing techniques,” which provided the dynamic modification capability required by the claims. EX1004, 8:11-14.

20. Since long before the ’641 patent, it has been the norm for a database-management system (DBMS) to allow simultaneous access to a database by multiple users at the same time, even if multiple users are trying to read and/or write) the same data item in the database.

21. Database-management systems use a form of concurrency control to enable simultaneous (“concurrent”) access of the same data by multiple users. Concurrency control has always been a core feature of databases and schemes for managing concurrent access to databases have existed dating back to development of databases themselves. For example, the multiversion concurrency control scheme (“MVCC”) was first proposed by Reed in 1976 in his Ph.D. thesis. EX1043. MVCC solves the problem of executing concurrent transactions on a database. *See* EX1041. Without concurrency control, concurrently executed transactions can cause undesirable results. EX1042, 465. For example, when a user performs an update (e.g., write) of a data item in a database and another user tries to retrieve (e.g., read) the same data item at the same time, the read and write transactions can interfere with each other. EX1041, 185.

22. MVCC solves the problem by creating multiple copies of each data item. EX1042, 466. The principle of MVCC algorithm is to create a multiversion database, e.g., each write on a data item produces a new copy (or version) of the data item, and for each read on the data item, the DBMS selects one of the versions of the data item to be read. EX1042, 466; EX1043.

23. MVCC was implemented in almost every commercially-available database management system available at the time of the '641 patent. *See* EX1045. Techniques for concurrency control in databases have been both so well understood and central to database design and implementation that virtually all database textbooks include a section, if not a chapter, on the topic. For example, textbooks I encountered as an undergraduate in the late 1970s and early 1980s such as “Principles of Database Systems” by Ullman,¹ first published in 1980, included a chapter on the topic. *See* EX1051. Current texts such as Ullman’s 2009 database text continue this practice, confirming the importance of the problem and the wide availability of known solutions. Therefore, the use of a standard commercially-available database-management system at the time of the '641 patent would have

¹ A distinguished computer scientist (Jeffrey D. Ullman) who would later go on to win the Turing Award, the computing profession’s version of the Nobel Prize in computer science.

enabled a database entries to be queried (i.e., read) while simultaneously being written to.

2. Exemplary commercial database-management system – Oracle 8i

24. Oracle Database version 8, also known as Oracle 8i, was released in June 1997. Oracle 8i Release 2 (8.1.6) was released in December 1999. Oracle resolves issues associated with concurrent access by using various types of locks and a multiversion consistency model. EX1044, 1-30.

Concurrency

A primary concern of a multiuser database management system is *how to control concurrency, or the simultaneous access of the same data by many users*.

...

Oracle resolves such issues by using various types of locks and a multiversion consistency model. Both features are discussed later in this section. These features are based on the concept of a transaction. It is the application designer's responsibility to ensure that transactions fully exploit these concurrency and consistency features.

...

Read consistency

Read consistency, as supported by Oracle, does the following:

...

- Ensures that *readers of database data do not wait for writers or other readers of the same data*

- Ensures that *writers of database data do not wait for readers of the same data[.] Ensures that writers only wait for other writers if they attempt to update identical rows in concurrent transactions.*

EX1044, 1-29, 1-30 (emphasis added).

25. Oracle 8i combines the “multiversion concurrency control and row-level locking” mechanism to solve the user contention of accessing the same row in a database. EX1044, 24-21. Oracle 8i introduces a “Row Lock[]” that only locks selected rows in a table of a database. EX1044, 24-20. Oracle 8i locks a row when the row is being updated (e.g., write transaction) to prevent another writer from modifying the same row. However, the Row Lock does not prevent simultaneous reading of the same row by another user. EX1044, 24-21.

Chapter 24, Data Concurrency and Consistency

This chapter explains how *Oracle provides concurrent access to and maintains the accuracy of shared information in a multiuser environment.* It describes the automatic mechanisms that Oracle uses to guarantee that the concurrent operations of multiple users do not interfere with each other.

EX1044, xxxiv (emphasis added).

Row Locks (TX)

The only DML locks Oracle acquires automatically are row-level locks. There is no limit to the number of row locks held by a statement or transaction, and Oracle does not escalate locks from the row level to a

coarser granularity. Row locking provides the finest grain locking possible and so provides the best possible concurrency and throughput.

The combination of multiversion concurrency control and row-level locking means that users contend for data only when accessing the same rows, specifically:

- *Readers of data do not wait for writers of the same data rows.*
- *Writers of data do not wait for readers of the same data rows* unless SELECT... FOR UPDATE is used, which specifically requests a lock for the reader.
- *Writers only wait for other writers if they attempt to update the same rows at the same time.*

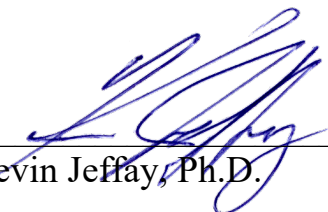
EX1044, 24-21 (emphasis added).

IV. CONCLUSION

26. In signing this declaration, I recognize that this declaration will be filed as evidence in an *inter partes* review before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in the case and that cross-examination will take place within the United States. If cross-examination is required, I will appear for cross-examination within the United States during the time allotted.

27. I hereby declare that all statements made herein are made of my own knowledge and are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Executed on this 17th day of May 2024 in Chapel Hill, NC.



Kevin Jeffay, Ph.D.