Trials@uspto.gov
571-272-7822

Paper 6
Entered: August 31, 2017

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SANDVINE CORPORATION and SANDVINE INCORPORATED ULC,
Petitioner,

v.

PACKET INTELLIGENCE, LLC,
Patent Owner.

Case IPR2017-00863
Patent 6,665,725 B1

Before ELENI MANTIS MERCADER, JUSTIN T. ARBES, and WILLIAM M. FINK, *Administrative Patent Judges*.

MANTIS MERCADER, *Administrative Patent Judge*.

DECISION
Institution of *Inter Partes* Review
37 C.F.R. § 42.108

I. INTRODUCTION

Sandvine Corporation and Sandvine Incorporated ULC (collectively, "Petitioner") filed a Petition for *inter partes* review of claims 1 and 2 of U.S. Patent No. 6,665,725 B1 (Ex. 1033, "the '725 patent"). Paper 1 ("Pet."). Patent Owner, Packet Intelligence, LLC, did not file a Preliminary Response. By statute, institution of an *inter partes* review may not be authorized "unless . . . the information presented in the petition . . . and any response . . . shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition." 35 U.S.C. § 314(a); *see also* 37 C.F.R. § 42.108.

Upon consideration of the Petition, we are persuaded Petitioner has demonstrated a reasonable likelihood of prevailing in establishing unpatentability of at least one claim of the '725 patent. Accordingly, we institute an *inter partes* review.

A. Related Matters

"Patent Owner submits that the '725 patent is the subject of a patent infringement lawsuit in the United States District Court for the Eastern District of Texas: *Packet Intelligence, LLC v. Sandvine Corp.*, Case No. 2:16-cv-00147, which was consolidated for pretrial matters (except venue) with co-pending *Packet Intelligence, LLC v. NetScout Systems, Inc.*, Case No. 2:16-cv-00230." Paper 4. Petitioner filed a petition for *inter partes* review challenging claims 10, 12, 13, and 15–17 of the '725 patent in IPR2017-00862. Petitioner also filed petitions for *inter partes* review of related United States Patent Nos. 6,839,751 B1 (IPR2017-00451); 6,771,646 B1 (IPR2017-00450); 6,954,789 B2 (IPR2017-00629 and IPR2017-00630); and 6,651,099 B1 (IPR2017-00769). *Id.*

B. The '725 Patent

The '725 patent relates to examining packets passing through a connection point on a computer network to determine whether a packet is of a conversational flow associated with an application program. Ex. 1033, 7:12–26. Figure 3 of the '725 patent is reproduced below.

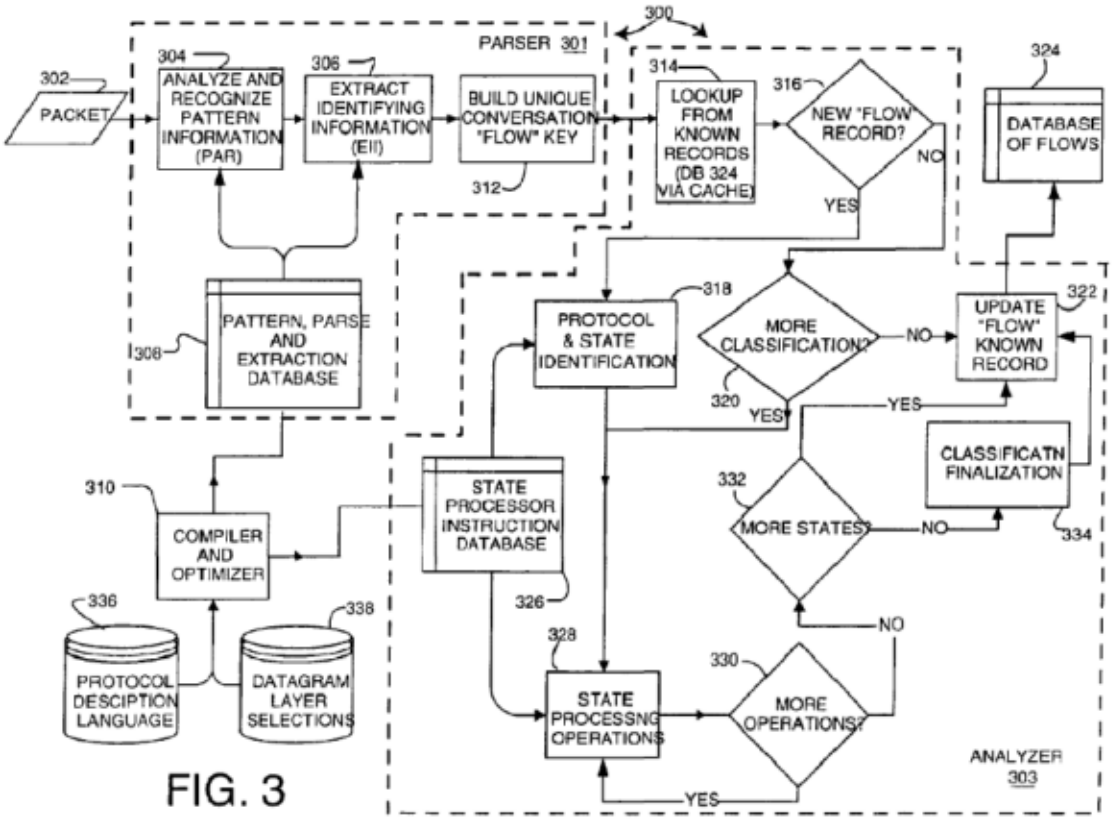


FIG. 3

Figure 3 above shows network packet monitor 300. *Id.* at 8:48–13:50.

Packet 302 is examined and evaluated by network packet monitor 300 to determine its characteristics, such as all the protocol information in a multilevel model, including what server application produced the packet. *Id.* at 8:51–57. Initialization of the monitor to generate what operations need to occur on packets of different types is accomplished by compiler and

optimizer 310, parsing and extraction of selected portions of packets to generate an identifying signature is accomplished by parser subsystem 301, and analysis of the packets is accomplished by analyzer 303. *Id.* at 8:64–9:3.

Parser subsystem 301 examines the packets using pattern recognition process 304, which parses the packet and determines the protocol types and associated headers for each protocol layer that exists in packet 302. *Id.* at 9:17–20. Protocol description language (PDL) files 336

> describe[] both patterns and states of all protocols that . . . occur at any layer, including how to interpret header information, how to determine from the packet header information the protocols at the next layer, and what information to extract for the purpose of identifying a flow, and ultimately, applications and services.

Id. at 9:29–35.

The ’725 patent states that it incorporates by reference U.S. Patent Application No. 09/608,237, issued as U.S. Patent 6,651,099 B1 (Ex. 1003, “the ’099 patent”), which discloses “protocol specific operations on individual packets including extracting information from header fields in the packet used for building a signature for identifying the conversational flow of the packet and for recognizing future packets as belonging to a previously encountered flow.” Ex. 1033, 2:21–30. A parser recognizes different patterns in the packet identifying the protocols used. *Id.* at 2:30–32. For each protocol recognized, packet elements are extracted to form the flow signature (also called a “key”). *Id.* at 2:32–34.

Compiler/optimizer 310 generates two sets of internal data structures. *Id.* at 9:42–43, Fig. 3. The first is the set of parsing/extraction operations 308 wherein “database 308 of parsing/extraction operations includes

information describing how to determine a set of one or more protocol dependent extraction operations from data in the packet that indicate a protocol used in the packet." *Id.* at 9:43–52. "The other internal data structure that is built by compiler 310 is the set of state patterns and processes 326." *Id.* at 9:53–54.

> These are the different states and state transitions that occur in different conversational flows, and the state operations that need to be performed (e.g., patterns that need to be examined and new signatures that need to be built) during any state of a conversational flow to further the task of analyzing a conversational flow.

Id. at 9:54–60.

Input to compiler/optimizer 310 "includes a set of files that describe each of the protocols that can occur." *Id.* at 41:24–25. "These files are in a convenient protocol description language (PDL) which is a high level language." *Id.* at 41:25–27. "The PDL file for a protocol provides the information needed by compilation process 310 to generate the database 308." *Id.* at 41:57–59.

> That database in turn tells [parser subsystem 301] how to parse and/or extract information, including one or more of what protocol-specific components of the packet to extract for the flow signature, how to use the components to build the flow signature, where in the packet to look for these components, where to look for any child protocols, and what child recognition patterns to look for.

Id. at 41:59–65

C. Illustrative Claim

Claim 1 of the challenged claims of the '725 patent is independent. Claim 1 is illustrative of the claimed subject matter:

1. A method of performing protocol specific operations on a packet passing through a connection point on a computer network, the method comprising:

(a) receiving the packet;

(b) receiving a set of protocol descriptions for a plurality of protocols that conform to a layered model, a protocol description for a particular protocol at a particular layer level including:

(i) if there is at least one child protocol of the protocol at the particular layer level, the-one or more child protocols of the particular protocol at the particular layer level, the packet including for any particular child protocol of the particular protocol at the particular layer level information at one or more locations in the packet related to the particular child protocol,

(ii) the one or more locations in the packet where information is stored related to any child protocol of the particular protocol, and

(iii) if there is at least one protocol specific operation to be performed on the packet for the particular protocol at the particular layer level, the one or more protocol specific operations to be performed on the packet for the particular protocol at the particular layer level: and

(c) performing the protocol specific operations on the packet specified by the set of protocol descriptions based on the base protocol of the packet and the children of the protocols used in the packet,

the method further comprising:

storing a database in a memory, the database generated from the set of protocol descriptions and including a data structure containing information on the possible protocols and

> organized for locating the child protocol related information for any protocol, the data structure contents indexed by a set of one or more indices, the database entry indexed by a particular set of index values including an indication of validity,
>
> wherein the child protocol related information includes a child recognition pattern,
>
> wherein step (c) of performing the protocol specific operations includes, at any particular protocol layer level starting from the base level, searching the packet at the particular protocol for the child field, the searching including indexing the data structure until a valid entry is found, and
>
> whereby the data structure is configured for rapid searches using the index set.

Ex. 1033, 95:2–49.

D. Reference

Petitioner relies on the following reference. Pet. 1.

Reference	**Title**	**Date**	**Ex. No.**
Baker	WO 97/23076 A1	June 26, 1997	Ex. 1038

E. Asserted Ground of Unpatentability

Petitioner contends that claims 1 and 2 of the '725 patent are unpatentable based on the following ground:

Reference	**Basis**	**Challenged Claims**
Baker	§ 102(b)	1 and 2

Pet. 1. Petitioner also relies on the declaration of Bill Lin, Ph.D. (Ex. 1006) for support. *Id.* at 2.

II. ANALYSIS

A. Claim Construction

In an *inter partes* review, claim terms in an unexpired patent are given their "broadest reasonable construction in light of the specification of the patent in which they appear." 37 C.F.R. § 42.100(b). Under the broadest reasonable construction standard, claim terms are given their ordinary and customary meaning, as would be understood by one of ordinary skill in the art in the context of the entire disclosure. *In re Translogic Tech., Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007). Only terms in controversy need to be construed, and only to the extent necessary to resolve the controversy. *Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999). Petitioner contends that the term "rapid" in claim 1 is not entitled to patentable weight because it is "purely functional" and a statement of a desired result, rather than an apparatus or specific structure to accomplish the desired result." Pet. 3 (citing *Hewlett-Packard Co. v. Bausch & Lomb Inc.*, 909 F.2d 1464, 1468 (Fed. Cir. 1990) ("[A]pparatus claims cover what a device is, not what a device does."); *In re Schreiber*, 128 F.3d 1473, 1478–79 (Fed. Cir. 1997) ("choosing to define an element functionally, i.e., by what it does, carries with it a risk"); *Euramax Int'l, Inc. v. Invisaflow*, LLC, Case IPR2016–00423, slip op. at 8–9 (PTAB June 1, 2016) (Paper 9). Petitioner asserts that affording no patentable weight to language describing an intended use or desired result of an apparatus is the long-standing rule. *Id.* at 3 (citing *In re Gardiner*, 171 F.2d 313, 315–16 (C.C.P.A. 1948) ("It is trite to state that the patentability of apparatus claims must be shown in the structure claimed and not merely upon a use, function, or result thereof.")).

We note Petitioner relies on legal precedents that address apparatus claims rather than method claims. Nonetheless, in *Hoffer v. Microsoft Corp.*, 405 F.3d 1326, 1329 (Fed. Cir. 2005), the court noted that a "whereby clause in a method claim is not given weight when it simply expresses the intended result of a process step positively recited" (quoting *Minton v. Nat'l Ass'n of Sec. Dealers, Inc.*, 336 F.3d 1373, 1381 (Fed. Cir. 2003)).

Here, the term "rapid" appears in the limitation "whereby the data structure is configured for rapid searches using the index set," which configuration appears to be the result of the organization of the data structure as recited by step (c). In particular, claim 1 recites, *inter alia*, "the data structure contents indexed by a set of one or more indices, the database entry indexed by a particular set of index values including an indication of validity, . . . wherein step (c) . . . includes . . . searching including indexing the data structure until a valid result is found." *See* Ex. 1033, 95:43–49. Similarly, the specification states that "the data structure is organized for rapidly locating the child protocol related information by using a set of one or more indices to index the contents of the data structure." *Id.* at 34:34–37. In other words, the intended result of organizing the data structure according to step (c) of the claim is a rapid search.

Accordingly, for purposes of this Decision, we agree with Petitioner that the term "rapid" is not entitled patentable weight.[1]

[1] The parties are encouraged to address the interpretation of the claim term "rapid" in their papers during trial.

B. Asserted Anticipation by Baker
1. Baker (Ex. 1038)

Figure 1 of Baker is reproduced below.

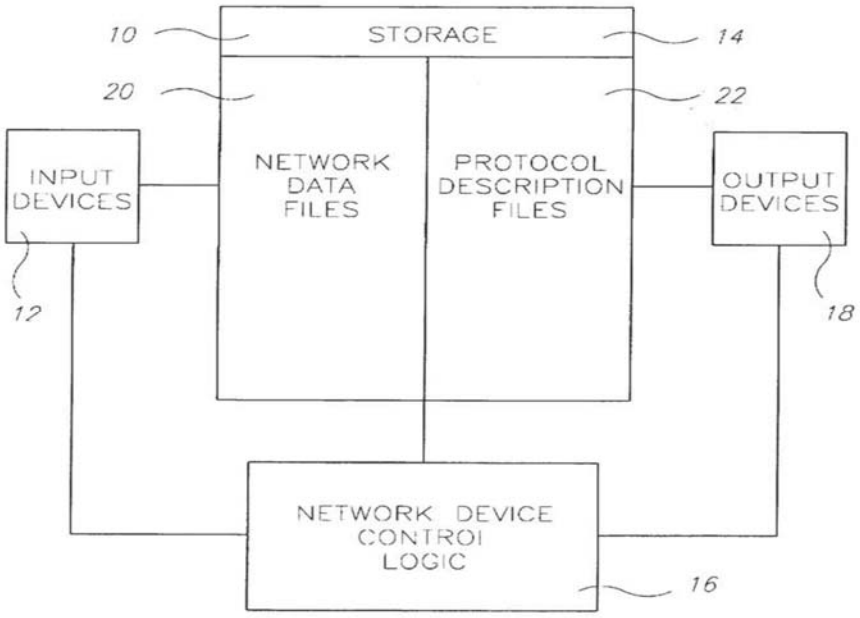


Figure 1 shows database of protocol description files 22 used by network device control logic 16 to retrieve network frames based on extracted field values and filtering criteria contained in protocol description files 22. Ex. 1038, 10:10–35.

Baker's protocol description file includes a protocol control record that defines the overall structure of a network protocol and references other information relating to the network protocol. *Id.* at 12:25–32. Each protocol description file includes a total bit length of the protocol header; a number of fields required to describe the header; and field records, each describing a protocol header field, including a byte offset from the start of the protocol header and, if appropriate, an associated lookup structure for determining the next protocol control record to use. *Id.* at 12:25–15:17, Tables 1, 2, and 4.

Upon initialization of the system, the protocol and associated control record information is extracted from all protocol description files and a "ProtocolList" is constructed. *Id.* at 20:35–21:11. The ProtocolList is a sorted vector of all protocol records. *Id.*

2. Analysis of Claim 1

Petitioner relies on Baker as allegedly teaching the limitations of independent claim 1. Pet. 8–43. Petitioner also relies upon the Declaration of Dr. Lin to support its contentions. Petitioner provided mappings of Baker's disclosure to claim 1 as further supported by Dr. Lin. *Id.* We emphasize Petitioner's contentions regarding each of the limitations of claim 1 below for which we are persuaded.

1(a) receiving the packet

Petitioner relies on Baker's disclosure that "*frames of network data may be received*" by a network device such as an analyzer for meeting the recited 1(a) limitation of "receiving the packet." Pet. 8 (citing Ex. 1038, 4:26–35).

1(b) receiving a set of protocol descriptions for a plurality of protocols that conform to a layered model

Petitioner points us to Baker's disclosure of protocol description files (PDFs), which is consistent with the '725 patent specification describing protocol description language files (PDLs) corresponding to one of a plurality of protocols (e.g., Ethernet, IP, TCP) conforming to a layered model according to Petitioner. Pet. 8–9 (citing Ex. 1033, 14:63–65, 42:14–22, 42:46–43:9; Ex. 1038, 21:32–22:5, 34:16–21). The existence of a layered model is further supported by Dr. Lin's testimony addressing Baker's disclosure of "ParseLvl" variable matched to a "[z]ero based

protocol level in ISO reference model of protocol being parsed (current protocol)" identifying the ISO reference model as the OSI layered model as it would have been recognized by a skilled artisan. *Id.* at 8–10 (citing Ex. 1006 ¶ 179 (discussing Table 14)); *see also* Ex. 1006 ¶¶ 30–37.

1(b)(i)(1)[2] *a protocol description for a particular protocol at a particular layer level including: (i) if there is at least one child protocol of the protocol at the particular layer level, the-one or more child protocols of the particular protocol at the particular layer level*

Petitioner relies on Baker's Ethernet PDF (i.e., "protocol description") for the Ethernet protocol (i.e., a particular protocol at a particular layer level) including one child protocol, Generic Protocol ("GP") (i.e., "including . . . the-one or more child protocols of the particular protocol at the particular layer level") for disclosing the claim limitation of "a protocol description for a particular protocol at a particular layer level including: (i) if there is at least one child protocol of the protocol at the particular layer level, the-one or more child protocols of the particular protocol at the particular layer level." Pet. 11–13 (citing Ex. 1038, Fig. 4D, 8:21–30, 21:32–22:5). Figure 4D of Baker is reproduced below.

Ethernet Type Lookup Structure

Protocol	Next Index	Minimum	Maximum	Mask	Translation
(None)	5	0x0000	0x8887	ALL	"Unknown"
Figure 5	5	0x8888	0x8888	ALL	"GP"
(None)	5	0x8889	0xFFFF	ALL	"Unknown"

Figure 4D shows the Ethernet type lookup structure including the one or more child protocols GP.

[2] For consistency purposes, we identify the claim limitations as Petitioner identified them, including adding parts (1) and (2) to element 1(b)(i).

Petitioner further directs us to the GP PDF (i.e., "protocol description") for the GP protocol (i.e., "a particular protocol at a particular layer level") including four child protocols, "GP1," "GP2," "GP3," and "GP4" (i.e., "the-one or more child protocols of the particular protocol at the particular layer level"). *Id.* at 13–14 (citing Ex. 1038, Figs. 5C–5E, 8:31–9:6, Table 13 (the Frame Type field indicates the "upper level protocol identifier," and Src Socket and Dst Socket fields indicate the "Socket of Upper-layer protocol"), Figs. 5, 5A (the Frame field references the Fig. 5C lookup structure, the Source Socket field references the Fig. 5D lookup structure, and the Destination Socket field references the Fig. 5E lookup structure). Figures 5C, 5D, and 5E of Baker are reproduced below.

Frame Type Next Protocol Structure					
Protocol	**Next Index**	**Minimum**	**Maximum**	**Mask**	**Translation**
[None]	4	0x00	0x00	ALL	"Illegal Protocol"
GP1	4	0x01	0x01	ALL	"GP1"
GP2	5	0x02	0x02	ALL	"GP2"
[None]	4	0x03	0xFF	ALL	"Illegal Protocol"

Figure 5C

Source Socket Next Protocol Structure					
Protocol	**Next Index**	**Minimum**	**Maximum**	**Mask**	**Translation**
[None]	8	0x0000	0x0142	ALL	"Unknown Protocol"
GP3	8	0x0143	0x018F	ODD	"GP3"
GP4	8	0x0143	0x018F	EVEN	"GP4"
[None]	8	0x0190	0xFFFF	ALL	"Illegal Protocol"

Figure 5D

Destination Socket Next Protocol Structure					
Protocol	**Next Index**	**Minimum**	**Maximum**	**Mask**	**Translation**
[None]	9	0x0000	0x0142	ALL	"Unknown Protocol"
GP3	9	0x0143	0x018F	ODD	"GP3"
GP4	9	0x0143	0x018F	EVEN	"GP4"
[None]	9	0x0190	0xFFFF	ALL	"Illegal Protocol"

Figure 5E

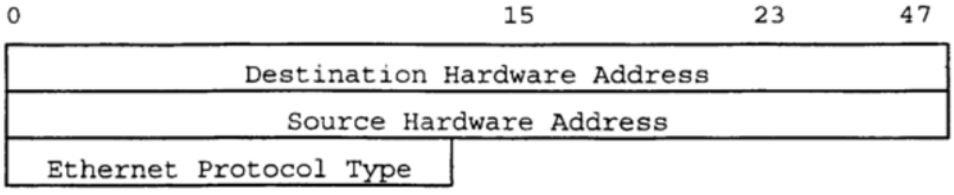
Figures 5C, 5D, and 5E show the GP PDF for GP, including four child protocols GP1, GP2, GP3, and GP4.

1(b)(i)(2) the packet including for any particular child protocol of the particular protocol at the particular layer level information at one or more locations in the packet related to the particular child protocol

Petitioner then directs us to Table 12 in Baker as disclosing the claim limitation of "the packet including for any particular child protocol of the particular protocol at the particular layer level information at one or more locations in the packet related to the particular child protocol." Pet. 14–18. Table 12 of Baker is reproduced below.

TABLE 12

ETHERNET v2.0 PROTOCOL SPECIFICATION



Destination Hardware Address - destination hardware station address (48 bits)

Source Hardware Address - source hardware station address (48 bits)

Ethernet Protocol Type - upper layer protocol designator (16 bits) 0x8888=GP

Table 12 shows the Ethernet Protocol Specification.

Specifically, according to Petitioner, Baker states that the "Ethernet Protocol Type" field of the Ethernet frame header includes an '"upper layer protocol designator 0x8888=GP' (i.e., information at one or more locations in the packet related to the particular child protocol)." *Id.* at 14–15 (citing Ex. 1038, Table 12, 21:14–16).

Petitioner contends that "each Ethernet frame includes an Ethernet header that is 14-bytes (112-bits) long followed by the header of the next layer protocol (GP Header) (i.e., information at a location in the packet related to the particular child protocol)." *Id.* at 15 (citing *id.* at 25:29–32 ("Frame 1 shown below has a hardware length of eighty-two 8-bit bytes and consists of a *fourteen byte Ethernet header*, a twenty byte GP header with no option bytes, and forty-eight bytes of application data.")). Frame 1 of Baker (Ex. 1038, 26) is reproduced below.

Frame (1)

(1)	08	00	00	00	00	03	08	00	00	00	00	04	88			
	88												Ethernet Header(14)			
(2)	35	00	00	44	B1	5F	00	01	08	00	01	47	01	02	03	04
	05	06	07	08									GP Header(20)			
(3)	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	00	00	00	00	00	00	00	00						Data (24)		
(4)	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
	00	00	00	00	00	00	00	00						Data (24)		

Frame 1 depicts an Ethernet frame.

1(b)(ii) the one or more locations in the packet where information is stored related to any child protocol of the particular protocol, and

As an example of the "one or more locations in the packet where information is stored," Petitioner discusses the "NumBits" attribute in Baker. Pet. 18–21. Each PDF protocol control record may include a "NumBits"

attribute that is "the total bit length of the protocol header." *Id.* at 18 (citing Ex. 1038, Table 1 ("numBits" attribute), 19:17; 56:30–57:11 (initialization routine for reading protocol control record elements from PDF, including "num_bits" attribute), Ex. 1006 ¶¶146–148). According to Petitioner, "[t]he '725 patent similarly discloses a 'HEADER' attribute of a PDL file 'used to describe the length of the protocol header.'" *Id.* at 18 (citing Ex. 1033, 48:41–50). According to Petitioner, "[d]uring the parsing process, the value of the numBits attribute may be used to determine where the next layer protocol header/data begins." *Id.* (referring to the analysis of element 1(c) discussed below). In pertinent part, in the analysis of element 1(c), Petitioner relies on Baker for disclosing that "[w]hen parsing of the current protocol terminates, the ParsePtr variable is set to the start of the next layer protocol header/data (i.e., by adding the value of ProtoParseLen (which for Ethernet is equal to the value of NumBits) to the ParsePtr variable)." *Id.* at 27 (citing Ex. 1038, 36:13–27, 64:11–13, 38:13–18, Figs. 4 ("NumBits" attribute), 13 (step 228); Ex. 1006 ¶¶174, 161, 165).

"Each PDF also includes field sub-records for each field of the protocol header" according to Petitioner. *Id.* at 19 (citing Ex. 1038, 12:25–28, Tables 1, 2). "Each field sub-record includes an 'fdwoff' attribute (also called Byte/Bit Offset) that is the byte offset of the field from the start of the protocol header." *Id.* (citing Ex. 1038, Table 2 ("fdwoff" attribute), 20:5, Figs. 4A, 5A, 147:28–148:13 (initialization routine for reading field sub-record elements from PDF, including "fdwoff" attribute); Ex. 1006 ¶¶ 149–150). The "fdwoff" attribute "is used during the parsing process to extract the value of the associated field." *Id.* (citing Ex. 1038, 27:17–35, Fig. 13 (step 210), Fig. 16 (step 502)).

Petitioner further relies on Baker's example of the "Ethernet PDF NumBits attribute of the protocol control record [being] set to 112, indicating the total bit length of the Ethernet header." *Id.* (citing Ex. 1038, Fig. 4). Figure 4 of Baker is reproduced below.

Ethernet Control Record					
Protocol Name	NumBits	NumFields	CurField	Fields	Options
Ethernet MAC Header	112	5	0	Figure 4a	(None)

Figure 4

Figure 4 indicates the Ethernet PDF NumBits attribute of 112.

Petitioner indicates that "[t]he Ethernet PDF 'Type' field sub-record includes a 'Bit Offset' (i.e., fdwoff attribute)," and shows "the Type field is 96-bits offset from the start of the Ethernet header." *Id.* (citing Ex. 1038, Fig. 4A; 29:32–30:4). Figure 4A of Baker is reproduced below.

Index	Field Name	Bit Offset	Bit Length	Left Shift	Right Shift	Check sum	Frame Length	Header Length	Statistics	Lookup Structure	Filter Structure	Format
0	Dst Vendor Address	0	24	0	16	0	0	0	(None)	Fig 4b	Fig 10.Idx 0	nex
1	Dst Station Address	24	24	0	16	0	0	0	(None)	(None)	Fig 10.Idx 1	nex
2	Src Vendor Address	48	24	0	16	0	0	0	(None)	Fig 4c	(None)	nex
3	Src Station Address	72	24	0	16	0	0	0	(None)	(None)	(None)	nex
4	Type	96	16	0	16	0	0	0	(None)	Fig 4d	(None)	nex

Figure 4A shows a "Bit Offset" of 96-bits offset.

Petitioner concludes that "the NumBits attribute of the Ethernet control record and the fdwoff attribute of the Ethernet Type field sub-record" constitute "the one or more locations in the packet where information is stored related to any child protocol of the particular protocol." *Id.* at 20.

1(b)(iii) if there is at least one protocol specific operation to be performed on the packet for the particular protocol at the particular layer level, the one

or more protocol specific operations to be performed on the packet for the particular protocol at the particular layer level; and

Petitioner contends "one or more protocol specific operations to be performed on the packet for the particular protocol at the particular layer level" as recited in claim 1 includes "one or more parsing and extraction operations on the packet," as recited in claim 10. Pet. 21 (quoting Ex. 1033, 42:3–6 ("[T]he PDL file for an Ethernet packet includes information on how the parsing subsystem is to extract the source and destination addresses, including where the locations and sizes of those addresses are."), citing *id.* at 41:57–65).

In particular, Petitioner states:

> The Baker PDFs include the locations and sizes of fields that are to be extracted by the parsing control logic (e.g., the disclosed ParseFrame, ParseProtocol, ParseFields, ValidateValue, and GetValue control logic), among other information. *See generally* EX1038 at 19:11-20:34, Tables, 1-13, Figs. 4, 4A-D, 5, 5A-D (describing data written to PDF file); 26:26-40:36, Figs. 11-16 (describing control logic), 64:1-68:24; EX1006, ¶¶157-177 (describing parsing control logic as implemented by disclosed source code). Namely, each PDF field sub-record includes an "fdwoff" attribute (also called Byte/Bit Offset) that is the byte offset of the respective field from start of protocol header (i.e., the location of the field that is to be extracted). EX1038 at Table 2 ("fdwoff" attribute), 20:4, 27:17-35, Figs. 4A, 5A, 13 (step 210), 16 (step 502), 147:28-148:13, 148:25-26; EX1006, ¶¶169, 185, 149-150; *see* Element 1(b)(ii). Each PDF field sub-record also includes an "fblen" attribute (also called Bit Length) that is the length of the respective field in bits (i.e., the size of the field that is to be extracted). EX1038 at Table 2 ("fblen" attribute), 20:4, 35:27-30, Figs. 4A, 5A, 13 (step 222), 147:28-148:13, 148:25-26; EX1006, ¶¶185, 149-150.

Pet. 22.

1(c) performing the protocol specific operations on the packet specified by the set of protocol descriptions based on the base protocol of the packet and the children of the protocols used in the packet

According to Petitioner, in the '725 patent, "the base protocol" of a packet is described as the protocol associated with the packet type, such as Ethernet. Pet. 23.

Petitioner contends Baker discloses "control logic perform[ing] protocol specific operations on the packet as specified by the set of protocol description files based on the base layer protocol (i.e., Ethernet) of the packet and the children of the protocols used in the packet (i.e., GP and others)." *Id.* at 23.

More particularly, Petitioner relies on Baker's disclosure that "after the Ethernet protocol fields are parsed (at step 106 of Fig. 11), 'the NextProtocol variable 'will refer to the GP shown in FIGS. 5-5(e), and ParsePtr will point at the start of line 2' of the frame.'" *Id.* at 27 (citing Ex. 1038, 38:13–18, 146:19–25, 67:27–32, 29:17–31, Figs. 11–14; Ex. 1006 ¶¶ 174, 161, 165). "[T]he CurrentProtocol variable will [then] be updated with the NextProtocol value (of GP) and the GP fields in the frame are parsed using the ParseProtocol, ParseFields, GetValue, and ValidateValue fields." *Id.* at 27–28 (citing Ex. 1038, 38:18–24, 67:27–32, Fig. 11 (steps 108, 128); Ex. 1006 ¶¶ 161–177).

1(d)(i) the method further comprising: storing a database in a memory, the database generated from the set of protocol descriptions and including a data structure containing information on the possible protocols and organized for locating the child protocol related information for any protocol

Petitioner contends that the recited "database" "may include a Protocol Table (PT) and a series of one or more Look Up Tables (LUTs)

associated with a protocol in PT and used to identify known protocols and their children." Pet. 29 (citing Ex. 1033, 14:39–43, 39:22–40:60, Fig. 18B). According to Petitioner, "Baker discloses a database in memory generated from the set of protocol descriptions consistent with the '725 Patent Fig. 18B embodiment." *Id.* (citing Ex. 1006 ¶¶ 182–187). In particular, "Baker discloses that, upon initialization and when PDF files are read into memory, the resulting protocol description data structures include a data structure of protocol records (each corresponding to a PDF) and further provides a ProtocolList sorted vector that is used as an index for searching the protocol descriptions supported by the system." *Id.* (citing Ex. 1038, 20:35–21:11, 127:20–128:13, 128:33–129:1; Ex. 1006 ¶¶ 143–147, 183).

Furthermore, Petitioner states that "[d]uring parsing, the protocol data structure is used to extract a field that specifies the next protocol, which is then used as an address into the lookup structure for determining the child protocol." *Id.* at 32 (citing Ex. 1038, 145:20–146:2, 64:18, 148:26, 27:17–35, Fig. 13 (step 210), Fig. 16, Table 2; Ex. 1006 ¶¶ 170–174, 186. Petitioner further states that "[w]hen the next protocol is identified using the lookup structure, it is used as an index into the protocol table." *Id.* at 33 (citing Ex. 1038, 146:19–25, 29:17–31, 67:27–32, Fig. 11 (steps 108, 128), Fig. 14 (steps 306, 308, 310, 312); Ex. 1006 ¶¶ 174, 161, 186). Petitioner contends that this is further supported by Dr. Lin's testimony. *Id.* (citing Ex. 1006 ¶¶ 182–188). Of note is Dr. Lin's testimony that the "Baker protocol data structure like the Protocol Table of the '725 patent, contains information on the possible protocols known by the system." Ex. 1006 ¶ 183 (comparing the '725 patent's protocol table (PT) with Baker's protocol data structure and ProtocolList).

1(d)(ii) the data structure contents indexed by a set of one or more indices

With respect to the claim 1 limitation of "the data structure contents indexed by a set of one or more indices," Petitioner contends Baker discloses that multi-protocol lookup structures are each indexed by a value extracted from the next/child protocol field in the packet. Pet. 33 (citing Ex. 1038, 177:4–26; 64:25–38, 175:35–176:24, 29:1–31, Fig. 13 (step 214), Fig. 14; Ex. 1006 ¶¶ 151–156, 170–172). Petitioner states "[t]he '725 patent similarly discloses that each LUT is 'indexed by one byte of the child recognition pattern that is extracted from the next protocol field in the packet.'" *Id.* (citing Ex. 1033, 39:33–35). Petitioner relies on Baker's disclosure of "[w]hen the next/child protocol is identified using the Baker lookup structure, it is used as an index into the protocol table." *Id.* (citing Ex. 1038, 146:19–25, 29:17–31, 67:27–32, Fig. 11 (steps 108, 128), Fig. 14 (steps 306, 308, 310, 312), Table 4; Ex. 1006 ¶¶ 174, 161, 190). According to Petitioner, this disclosure is consistent with the '725 patent wherein "when a lookup results in a valid next protocol, the next protocol 'is used as an index into the protocol table.'" *Id.* at 34 (citing Ex. 1033, 40:27–31).

1(d)(iii) the database entry indexed by a particular set of index values including an indication of validity

Petitioner argues that, as described in the '725 patent, "each LUT entry includes a 'node code' that indicates the validity of the contents." Pet. 34 (citing Ex. 1033, 39:39–50, 35:4–11, 14:57–61). The '725 patent discloses that "a 'protocol' node code indicates a recognized protocol whereas a 'null' node code indicates 'that there is no valid entry.'" *Id.*

Petitioner points us to Baker's "multi-protocol lookup structure entries," which "are indexed by a valid protocol identification value." *Id.* at

35 (citing Ex. 1006 ¶¶ 151–156, 170–172, 194). Petitioner adds that the "[e]ntries also include a 'prot' variable ('Protocol' of Table 4 and Figs. 4D, 5C–5E) that, similar to the '725 Patent 'node code,' has one of two values: (1) a pointer to a protocol record, indicating the protocol that has been recognized as the next/child protocol; or (2) a 'null' code indicating an 'unknown' or 'illegal' entry (i.e., there is no valid entry)." *Id.* (citing Ex. 1038, Figs. 5C–5D, Fig. 5E, Table 4 (describing "Protocol" variable as "pointer to protocol description structure"); Ex. 1006 ¶ 195).

1(d)(iv) wherein the child protocol related information includes a child recognition pattern

Petitioner contends an example of the recited "child recognition pattern" is depicted in Figure 5C of Baker, which "discloses that the child protocol (i.e., GP1) of a particular protocol (i.e., GP), is indicated by a child recognition pattern (i.e., 0x01(hex))." Pet. 39. Dr. Lin's testimony supports that Baker discloses a child recognition pattern. Dr. Lin testifies that "Baker discloses that the child protocol, e.g., GP1, of a particular protocol, e.g., GP, is indicated by a child recognition pattern, e.g., 0x01 (hex) at a location within the header of the particular protocol." Ex. 1006 ¶ 200.

1(e) wherein step (c) of performing the protocol specific operations includes, at any particular protocol layer level starting from the base level, searching the packet at the particular protocol for the child field, the searching including indexing the data structure until a valid entry is found

According to Petitioner, **"**Baker discloses ParseFields control logic, that iterates through fields of a packet that are defined in an associated indexed field array of a particular protocol description record in accordance with [Figure] 13." Pet. 40 (citing Ex. 1038, Fig. 13; Ex. 1006 ¶¶ 166–168). "For each field that is to be parsed in accordance with the protocol

description and starting at index (i=0) of the field sub-records array, the ParseFields control logic initiates the get_value function (also called the GetValue control logic) to determine the location of the field in the packet and extract a value from the field." *Id.* (citing Ex. 1038, 64:11–18; Ex. 1006 ¶¶ 168–169). "If the field includes a child recognition pattern, . . . Baker initiates the value_ok function (also called the ValidateValue control logic) to use the extracted value as an index to the associated lookup array and determine if the value is for a valid protocol." *Id.* (citing the analysis of claim element 1(d)(iii) discussed above). "If the currently parsed field [does not] contain a child recognition pattern or the value_ok function determines that there is not a valid entry for the child recognition pattern, the index used to increment through the field array is incremented by 1 and the ParseFields logic continues by parsing the next field of the field array." *Id.* at 40–41 (citing Ex. 1038, 146:9–18, 64:11–13, 29:7–16, Figs. 13, 14; Ex. 1006 ¶¶ 170–172).

If at any point, Petitioner contends, "the currently parsed field includes a child recognition pattern and the value_ok function determines that it is associated with a valid child protocol, the NextProtocol variable is set to the associated protocol description that is used to parse the next layer of the packet for the 'first valid field parsed in a protocol that specifies the NextProtocol.'" *Id.* at 41 (citing Ex. 1038, 146:19–25, 29:17–31, Fig. 14; Ex. 1006 ¶¶ 172–174). "The ParseFields logic continues implementing the get_value and value_ok functions until a valid child protocol is identified or when processing otherwise is terminated, at which point, parsing of the next layer protocol continues in accordance with the identified NextProtocol

protocol description using the ParseFields logic." *Id.* (citing Ex. 1038, 64:11–13, 36:13-17, Figs. 11, 13; Ex. 1006 ¶¶ 176, 168–175, 161–165).

Petitioner contends "Baker uses an extracted child recognition pattern as an index to the associated lookup array and determines if the child recognition pattern is valid." *Id.* at 42 (citing Ex. 1006 ¶¶ 170–172, 190, 206; claim element 1(d)(iv) analysis discussed above). "Searching (i.e., incrementing the field array index, extracting value, determining validity) continues until a valid child recognition pattern is found." *Id.* (citing Ex. 1006 ¶ 206).

1(f) and whereby the data structure is configured for rapid searches using the index set

As discussed above, at this stage of the proceeding and based on the current record, we agree with Petitioner's position that the data structure being configured for "rapid" searches is not entitled to patentable weight. *See supra* Section II.A.

However, we also agree with Petitioner based on the current record that if "rapid" searches in claim 1 is deemed to be limiting, a person of ordinary skill in the art would have understood that Baker discloses the same within the meaning of the '725 patent. Pet. 43 (citing Ex. 1006 ¶¶ 209–210). More specifically, the '725 patent discloses "rapid searches using the index set" in the following context:

> An alternate embodiment of the data structure used in database 308 is illustrated in FIG. 18B. [T]he data structure permits rapid searches to be performed by the pattern recognition process 304 by indexing locations in a memory rather than performing address link computations. . . . The pattern matching is carried out by finding particular "child recognition codes" in the header fields, and using these codes to index one or more of the LUT's.

Id. (citing Ex. 1033, 14:33–62) (emphasis omitted).

Petitioner contends Baker discloses that the "data structure uses a child recognition code/pattern to index multi-protocol next protocol lookup arrays in the exact same way." *Id.* (citing Ex. 1038, 177:4–26; Ex. 1006 ¶ 210). "Additionally, searching the Baker multi-protocol lookup arrays is performed by indexing locations in memory rather than performing address link computations." *Id.* (citing Ex. 1038, 64:25–38, 175:35–176:24, 29:1–31, Fig. 13 (step 214), Fig. 14; Ex. 1006 ¶¶ 170–172, 210). "Searching the Baker multi-protocol lookup arrays is the same as searching the Lookup Tables of the '725 [p]atent." *Id.* (comparing Baker, as discussed above, with Ex. 1033, 14:33–62; citing Ex. 1006 ¶¶ 209–210).

Thus, according to Petitioner, "searching the Baker multi-protocol lookup arrays by indexing locations in memory" constitutes "a 'rapid search' using the child recognition pattern index set within the meaning of the '725 patent." *Id.* (citing Ex. 1006 ¶¶ 209–210).

We have reviewed Petitioner's analysis and the supporting testimony of Dr. Lin, which, at this time, are unrebutted by Patent Owner. On this record, for the foregoing reasons, we are persuaded Petitioner is reasonably likely to prevail on the asserted ground of anticipation against claim 1.

3. Claim 2

Claim 2 depends from claim 1. Petitioner asserts a challenge to claim 2. Pet. 44–46. We have reviewed Petitioner's mapping of Baker to claim 2 and supporting testimony of Dr. Lin, which, at this time, are unrebutted by Patent Owner. On this record, we determine Petitioner has sufficiently shown how Baker discloses the respective limitations of claim 2. Accordingly, we determine Petitioner has established a reasonable likelihood

of prevailing on the asserted ground of anticipation under § 102(b) based on Baker.

III. CONCLUSION

We conclude that Petitioner has demonstrated a reasonable likelihood of prevailing with respect to at least one claim of the '725 patent challenged in the Petition. Therefore, we institute an *inter partes* review on the sole asserted ground as to the challenged claims. At this stage of the proceeding, the Board has not made a final determination with respect to the patentability of the challenged claims.

IV. ORDER

In consideration of the foregoing, it is hereby:

ORDERED that an *inter partes* review is instituted as to claims 1 and 2 on the following ground:

Claims 1 and 2 as anticipated by Baker under 35 U.S.C. § 102(b);

FURTHER ORDERED that pursuant to 35 U.S.C. § 314(a), *inter partes* review of the '725 patent is instituted with trial commencing on the entry date of this Order, and pursuant to 35 U.S.C. § 314(c) and 37 C.F.R. § 42.4, notice is given of the institution of the trial; and

FURTHER ORDERED that the trial is limited to the ground identified immediately above, and no other ground is authorized.

PETITIONER:

Eric A. Buresh
Mark C. Lang
Kathleen D. Fitterling
Abran J. Kean
ERISE IP, P.A.
eric.buresh@eriseip.com
mark.lang@eriseip.com
kathleen.fitterling@eriseip.com
abran.kean@eriseip.com

PATENT OWNER:

Steven W. Hartsell
Alexander E. Gasser
Paul J. Skiermont
Sadaf R. Abdullah
SKIERMONT DERBY LLP
shartsell@skiermontderby.com
agasser@skiermontderby.com
packetintelligence@skiermontderby.com