

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

JUNIPER NETWORKS, INC.

Petitioner,

v.

ORCKIT CORPORATION

Patent Owner.

Case IPR2024-00026

U.S. Patent No. 7,545,740

**PETITION FOR *INTER PARTES* REVIEW OF
U.S. PATENT NO. 7,545,740**

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	STANDING, IDENTIFICATION OF CHALLENGE, AND RELIEF REQUESTED	1
	A. Grounds for Standing (§42.104(a)).....	1
	B. Identification of Challenge and Relief Requested (§42.104(b)).....	1
III.	TECHNOLOGY BACKGROUND	4
	A. Link Aggregation.....	4
	B. Multiplexing and Demultiplexing.....	5
IV.	SUMMARY OF THE '740 PATENT	8
	A. Overview of the '740 Patent	8
	B. Prosecution History.....	12
	C. Level of Ordinary Skill in the Art.....	13
	D. Claim Construction.....	14
V.	DISCRETIONARY DENIAL ISN'T APPROPRIATE	16
	A. Discretionary Denial Under the <i>Fintiv</i> factors Isn't Appropriate	16
	B. Discretionary Denial Under <i>General Plastic</i> Isn't Appropriate	17
VI.	GROUND 1 AND 2: BRUCKMAN ALONE OR IN VIEW OF BASS RENDER OBVIOUS CLAIMS 1-31.	18
	A. Bruckman Summary	18
	B. Basso Summary	22
	C. Reasons to Combine Bruckman and Basso	24

D.	Holdsworth Summary	27
E.	Holdsworth Shows Background Knowledge of a POSITA.....	28
F.	Claim 1.....	29
G.	Claim 2.....	42
H.	Claim 3.....	45
I.	Claim 4.....	45
J.	Claim 5.....	46
K.	Claim 6.....	47
L.	Claim 7.....	48
M.	Claim 8.....	49
N.	Claim 9.....	50
O.	Claim 10.....	51
P.	Claim 11.....	54
Q.	Claim 12.....	59
R.	Claim 13.....	59
S.	Claim 14.....	61
T.	Claim 15.....	63
U.	Claim 16.....	64
V.	Claim 17.....	64
W.	Claim 18.....	67
X.	Claim 19.....	67

Y.	Claim 20.....	67
Z.	Claims 21-27.....	68
AA.	Claim 28.....	68
BB.	Claim 29.....	69
CC.	Claim 30.....	69
DD.	Claim 31.....	70
VII.	GROUNDS 3 AND 4: BRUCKMAN (ALONE OR WITH BASSO) IN VIEW OF HOLDSWORTH RENDER OBVIOUS CLAIMS 11 AND 26..	70
A.	Reasons to Combine Holdsworth with Bruckman.....	70
B.	Claims 11, 26.....	74
VIII.	CONCLUSION.....	74

EXHIBIT LIST

Exhibit	Description
1001	U.S. Patent No. 7,545,740 to Zelig <i>et al.</i>
1002	Prosecution History of U.S. 7,545,740
1003	Declaration of Dr. Henry Houh
1004	Curriculum Vitae of Dr. Houh
1005	U.S. Patent Publication No. 2004/0228278 to Bruckman <i>et al.</i> (“Bruckman”)
1006	U.S. Patent Publication No. 2003/0210688 to Basso <i>et al.</i> (“Basso”)
1007	Digital Logic Design, by Holdsworth (2002) (“Holdsworth”)
1008	U.S. Patent No. 6,943,580 to Lewis
1009	IEEE Standard Dictionary of Electrical and Electronics Terms (1984)
1010	RFC 1042, A Standard for the Transmission of IP Datagrams over IEEE 802 Networks
1011	C++ Inside & Out, by Bruce Eckel (1993)
1012	Complaint, <i>Orckit Corporation v. Cisco Systems, Inc.</i> , No. 2:22-cv-276-JRG-RSP (E.D. Tex., Jul. 22, 2022)
1013	Federal Court Statistics

Exhibit	Description
1014	Proposed Docket Control Order, <i>Orckit Corporation v. Cisco Systems, Inc.</i> , No. 2:22-cv-276-JRG-RSP (E.D. Tex., Dec. 2, 2022)
1015	Infringement Contentions, <i>Orckit Corporation v. Cisco Systems, Inc.</i> , No. 2:22-cv-276-JRG-RSP (E.D. Tex., Nov. 3, 2022)
1016	Affidavit from the Internet Archive
1017	U.S. Patent No. 7,271,993 to Plojhar
1018	RFC 793, Transmission Control Protocol
1019	RFC 2026, The Internet Standards Process – Revision 3
1020	U.S. Patent No. 5,960,428 to Lindsay <i>et al.</i>
1021	U.S. Patent No. 5,999,538 to Haddock <i>et al.</i>
1022	Course Catalog for Electrical Engineering
1023	Copyright Record for Eckel (Ex.1011)
1024	U.S. Patent No. 6,473,424 to DeJager

CLAIM LISTING

Claims 1-31	
1.0	A method for communication, comprising:
1.1	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel,
1.2	at least one of said first physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction;
1.3	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel,
1.4	at least one of said second physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction;
1.5	receiving a data frame having frame attributes sent between the communication network and the network node;
1.6	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and
1.7	sending the data frame over the selected first and second physical links, said sending comprising communicating along at least one of said bi-directional links.
2.1	The method according to claim 1, wherein the network node comprises a user node, and
2.2	wherein sending the data frame comprises establishing a communication service between the user node and the communication network.
3.1	The method according to claim 1, wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled.
4.0	A method for communication, comprising:
4.1	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;
4.2	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;
4.3	receiving a data frame having frame attributes sent between the communication network and the network node;

4.4	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and
4.5	sending the data frame over the selected first and second physical links,
4.6	at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.
5.0	A method for communication, comprising:
5.1	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;
5.2	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;
5.3	receiving a data frame having frame attributes sent between the communication network and the network node;
5.4	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and
5.5	sending the data frame over the selected first and second physical links,
5.6	coupling the network node to the one or more interface modules comprises aggregating two or more of the first physical links into an external Ethernet link aggregation (LAG) group
5.7	so as to increase a data bandwidth provided to the network node.
6.1	The method according to claim 1, wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network, and
6.2	demultiplexing downstream data frames sent from the communication network to the network node.
7.1	The method according to claim 1, wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links.
8.1	The method according to claim 1, wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes.
9.1	The method according to claim 8, wherein applying the mapping function comprises applying a hashing function.
10.1	The method according to claim 9, wherein applying the hashing function comprises determining a hashing size responsively to a number of at least some of the first and second physical links,

10.2	applying the hashing function to the at least one of the frame attributes to produce a hashing key,
10.3	calculating a modulo of a division operation of the hashing key by the hashing size, and
10.4	selecting the first and second physical links responsively to the modulo.
11.1	The method according to claim 10, wherein selecting the first and second physical links responsively to the modulo comprises selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.
12.1	The method according to claim 1, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.
13.0	A method for communication, comprising:
13.1	coupling a network node to one or more interface modules using a first group of first physical links arranged in parallel;
13.2	coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel;
13.3	receiving a data frame having frame attributes sent between the communication network and the network node;
13.4	selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group; and
13.5	sending the data frame over the selected first and second physical links,
13.6	coupling the network node to the one or more interface modules and coupling each of the one or more interface modules to the communication network comprising specifying bandwidth requirements comprising at least one of a committed information rate (CIR), a peak information rate (PIR) and an excess information rate (EIR) of a communication service provided by the communication network to the network node, and
13.7	allocating a bandwidth for the communication service over the first and second physical links responsively to the bandwidth requirements.
14.0	A method for connecting user ports to a communication network, comprising:
14.1	coupling the user ports to one or more user interface modules;

14.2	coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel,
14.3	at least one of said backplane traces being bi-directional and operative to communicate in both an upstream direction and a downstream direction;
14.4	receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes;
14.5	for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces; and
14.6	sending the data frame over the selected backplane trace; said sending comprising communicating along said at least one of said backplane traces.
15.0	A method for connecting user ports to a communication network, comprising:
15.1	coupling the user ports to one or more user interface modules;
15.2	coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel;
15.3	receiving data frames sent between the user ports and the communication network, the data frames having respective frame attributes;
15.4	for each data frame, selecting responsively to at least one of the respective frame attributes a backplane trace from the two or more backplane traces; and
15.5	sending the data frame over the selected backplane trace, at least some of the backplane traces being aggregated into an Ethernet link aggregation (LAG) group.
16.1	The method according to claim 14, wherein selecting the backplane trace comprises applying a hashing function to the at least one of the frame attributes.
17.0	Apparatus for connecting a network node with a communication network, comprising:
17.1	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network,
17.2	at least one of said interface modules being operative to communicate in both an upstream direction and a downstream direction;

17.3	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;
17.4	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and
17.5	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame;
17.6	at least one of said first physical links and at least one of said second links being bi-directional links operative to communicate in both said upstream direction and said downstream direction.
18.1	The apparatus according to claim 17, and comprising a backplane to which the one or more interface modules are coupled, wherein the second physical links comprise backplane traces formed on the backplane.
19.0	Apparatus for connecting a network node with a communication network, comprising:
19.1	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;
19.2	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;
19.3	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and
19.4	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,
19.5	at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group.
20.0	Apparatus for connecting a network node with a communication network, comprising:
20.1	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;

20.2	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;
20.3	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and
20.4	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,
20.5	two or more of the first physical links being aggregated into an external Ethernet link aggregation (LAG) group so as to increase a data bandwidth provided to the network node.
21.1	The apparatus according to claim 17, and comprising a multiplexer, which is arranged to perform at least one of multiplexing upstream data frames sent from the network node to the communication network, and
21.2	demultiplexing downstream data frames sent from the communication network to the network node.
22.1	The apparatus according to claim 17, wherein the control module is arranged to balance a frame data rate among at least some of the first and second physical links.
23.1	The apparatus according to claim 17, wherein the control module is arranged to apply a mapping function to the at least one of the frame attributes so as to select the first and second physical links.
24.1	The apparatus according to claim 23, wherein the mapping function comprises a hashing function.
25.1	The apparatus according to claim 24, wherein the control module is arranged to determine a hashing size responsively to a number of at least some of the first and second physical links,
25.2	to apply the hashing function to the at least one of the frame attributes to produce a hashing key,
25.3	to calculate a modulo of a division operation of the hashing key by the hashing size, and
25.4	to select the first and second physical links responsively to the modulo.
26.1	The apparatus according to claim 25, wherein the control module is arranged to select the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo.

27.1	The apparatus according to claim 17, wherein the at least one of the frame attributes comprises at least one of a layer 2 header field, a layer 3 header field, a layer 4 header field, a source Internet Protocol (IP) address, a destination IP address, a source medium access control (MAC) address, a destination MAC address, a source Transmission Control Protocol (TCP) port and a destination TCP port.
28.0	Apparatus for connecting a network node with a communication network, comprising:
28.1	one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network;
28.2	a first group of first physical links arranged in parallel so as to couple the network node to the one or more interface modules;
28.3	a second group of second physical links arranged in parallel so as to couple the one or more interface modules to the communication network; and
28.4	a control module, which is arranged to select for each data frame sent between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame,
28.5	the communication network being arranged to provide a communication service to the network node, the service having specified bandwidth requirements comprising at least one of a committed information rate (CR), a peak information rate (PIR) and an excess information rate (EIR), and
28.6	the first and second groups of physical links being dimensioned to provide an allocated bandwidth for the communication service responsively to the bandwidth requirements.
29.0	Apparatus for connecting user ports to a communication network, comprising:
29.1	one or more user interface modules coupled to the user ports, which are arranged to process data frames having frame attributes sent between the user ports and the communication network,
29.2	at least one of said user interface modules being bi-directional and operative to communicate in both an upstream direction and a downstream direction;
29.3	a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in

	parallel so as to transfer the data frames between the one or more user interface modules and the communication network,
29.4	at least one of said backplane traces being bi-directional and operative to communicate in both said upstream direction and said downstream direction; and
29.5	a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame.
30.0	Apparatus for connecting user ports to a communication network, comprising:
30.1	one or more user interface modules coupled to the user ports,
30.2	which are arranged to process data frames having frame attributes sent between the user ports and the communication network;
30.3	a backplane having the one or more user interface modules coupled thereto and comprising a plurality of backplane traces arranged in parallel so as to transfer the data frames between the one or more user interface modules and the communication network;
30.4	a control module, which is arranged to select, for each data frame, responsively to at least one of the frame attributes, a backplane trace from the plurality of backplane traces over which to send the data frame;
30.5	at least some of the backplane traces are aggregated into an Ethernet link aggregation (LAG) group.
31.1	The apparatus according to claim 29, wherein the control module is arranged to apply a hashing function to the at least one of the frame attributes so as to select the backplane trace.

MANDATORY NOTICES UNDER 37 C.F.R. §42.8(a)(1)

- **§42.8(b)(1):** Juniper Networks, Inc. is the real party-in-interest. No other party funded, controlled, and/or directed this filing.
- **§42.8(b)(2):** The '740 Patent is at issue in the following cases that may affect, or be affected by, a decision in this proceeding:

Orckit Corporation v. Juniper Networks, Inc., No. 1:23-cv-00822 (D. Del.);

Orckit Corporation v. Cisco Systems, Inc., No. 2:22-cv-00276 (E.D. Tex.);

Orckit Corporation v. Arista Networks, Inc., No. 1:23-cv-00821 (D. Del.); and

Cisco Systems, Inc. v. Orckit Corporation, IPR2023-00401 (P.T.A.B.). A

Motion for Joinder accompanies this Petition which presents grounds identical to those presented in IPR2023-00401.

- **§§42.8(b)(3), (4):** A Power of Attorney accompanies this Petition. Juniper consents to email service at ipr@fischllp.com.

Lead Counsel	Back-up Counsel
Kyle K. Tsui (Reg. No. 62,602) kyle.tsui@fischllp.com FISCH SIGLER LLP 5301 Wisconsin Avenue NW Suite 400 Washington, DC 20015 Phone: 202.362.3527	Andrew L. Ramos (Reg. No. 76,865) andrew.ramos@fischllp.com FISCH SIGLER LLP 5301 Wisconsin Avenue NW Suite 400 Washington, DC 20015 Phone: 202.362.3522

I. INTRODUCTION

U.S. Patent No. 7,545,740 (“the ’740 Patent”) relates to link aggregation which involves joining a group of parallel physical links between two endpoints together into a single logical link. To gain allowance, the Applicant amended the claims to recite that at least one of the links in a link aggregation group was “a bi-directional link operative to communicate in both an upstream direction and a downstream direction.”¹ But as shown below, and as Dr. Houh’s Declaration (Ex. 1003) confirms, the concept of bi-directional aggregated links was known and would’ve been obvious to a person of ordinary skill in the art.

Included with this Petition is a timely Motion for Joinder. Juniper Networks, Inc. (“Juniper,” “Petitioner”) respectfully requests *inter partes* review (“IPR”) of claims 1-31 (“the Challenged Claims”) of the ’740 Patent.

II. STANDING, IDENTIFICATION OF CHALLENGE, AND RELIEF REQUESTED

A. Grounds for Standing (§42.104(a))

Juniper certifies the ’740 Patent is available for IPR. Juniper isn’t barred or estopped from requesting IPR of the Challenged Claims on the asserted grounds.

B. Identification of Challenge and Relief Requested (§42.104(b))

A Motion for Joinder accompanies this Petition, which, in the interest of judicial economy, relies on copies of the exhibits filed in IPR2023-00401. Relying

¹ Ex. 1002, 59.

on this identical evidence challenging the same claims with substantially identical grounds, Juniper asks the Patent Trial and Appeal Board (“the Board”) to find the Challenged Claims unpatentable under §103² based on the following patents and printed publications in view of the general knowledge of a person having ordinary skill in the art (“POSITA”) prior to the purported invention:

Ground	Proposed Challenges
1	Bruckman ³ renders obvious claims 1-31.
2	Bruckman and Basso ⁴ render obvious claims 1-31.
3	Bruckman and Holdsworth ⁵ render obvious claims 11 and 26.
4	Bruckman, Bass, and Holdsworth render obvious claims 11 and 26.

Bruckman published on November 18, 2004. Basso published on November 13, 2003. And Holdsworth is a textbook having ISBN 0-7506-45882 and a 2002 copyright date. Holdsworth was published in 2001 and on sale as of at least August 21, 2001.⁶ U.S. Patent No. 7,271,993 cites this edition of Holdsworth, confirming its availability and a POSITA’s knowledge of it.⁷ The totality of the evidence

² 35 U.S.C. §§102, 103 (pre-AIA) applies to the ’740 Patent.

³ U.S. Patent Publication No. 2004/0228278 (Ex. 1005) (“Bruckman”).

⁴ U.S. Patent Publication No. 2003/0210688 (Ex. 1006) (“Basso”).

⁵ Logic Design, by Holdsworth (Ex. 1007) (“Holdsworth”).

⁶ See Ex. 1016 (screenshots from the Internet Archive).

⁷ Ex. 1017, 6:5-7 (citing Holdsworth); Houh, ¶56.

demonstrates Holdsworth was a printed publication by 2005.⁸ Bruckman, Basso, and Holdsworth are therefore each prior art under §102(b).

Petitioner also cites additional prior art as evidence of a POSITA's background knowledge and contemporaneous context supporting Petitioner's assertions regarding what a POSITA would've understood from the prior art.⁹ For example, Dr. Houh and this Petition cite to "Request for Comments" ("RFC") documents published by the Internet Engineering Task Force ("IETF"). The Board has repeatedly found IETF documents, including RFCs, qualify as "printed publications" because they're "precisely the type of documents whose main purpose is for public disclosure."¹⁰

⁸ See *Hulu, LLC v. Sound View Innovations, LLC*, IPR2018-01039, Paper 29 at 17-18 (P.T.A.B. Dec. 20, 2018) (precedential) (considering the "totality of the evidence" when assessing whether a reference is prior art).

⁹ See *Yeda Research v. Mylan Pharm. Inc.*, 906 F.3d 1031, 1041-1042 (Fed. Cir. 2018) (affirming the use of "supporting evidence relied upon to support the challenge"); see also *K/S HIMPP v. Hear-Wear Techs., LLC*, 751 F.3d 1362, 1365-66 (Fed. Cir. 2014); *Arendi S.A.R.L. v. Apple Inc.*, 832 F.3d 1355, 1363 (Fed. Cir. 2016); 37 C.F.R. §42.104(b).

¹⁰ *Apple, Inc. v. VirnetX, Inc.*, IPR2017-00337, Paper 31 at 46-47 (P.T.A.B. May 30, 2018); see also, e.g., *Riot Games, Inc. v. Paltalk Holdings, Inc.*, IPR2018-00130, Paper 11 at 30-33 (P.T.A.B. May 15, 2018) (RFCs are printed publications).

III. TECHNOLOGY BACKGROUND

A. Link Aggregation

Link aggregation “is a technique by which a group of parallel physical links between two endpoints in a data network can be joined together into a single logical link.”¹¹ “Link aggregation offers benefits of increased bandwidth, as well as increased availability, since the logical link can continue to function (possibly with reduced bandwidth) even when one of the physical links fails or is taken out of service.”¹² Link aggregation for Ethernet networks is defined by industry standards.¹³

Link aggregation includes a process for selecting which of the physical links in a Link Aggregation Group (LAG) will transmit a particular frame. This process employs “a distributor function, which distributes data frames submitted by MAC clients among the physical links in the group, and a collector function, which receives frames over the aggregated links and passes them to the appropriate MAC clients.”¹⁴ Industry standards describe “possible distribution algorithms that meet the requirements of the standard, while providing some measure of load balancing

¹¹ Bruckman, [0002].

¹² *Id.*, [0002].

¹³ *See id.*, [0003].

¹⁴ *Id.*, [0003].

among the physical links in the aggregation group.”¹⁵ These distribution algorithms commonly use a hash function.¹⁶

Hashing techniques used for distributing packets or data frames across link aggregation configurations were well-known before the '740 Patent was filed.¹⁷

B. Multiplexing and Demultiplexing

Link aggregation uses a concept referred to as multiplexing. A multiplexer selects from a plurality of input lines for transmission over a single output line. An example 4-input multiplexer implementation is shown below:

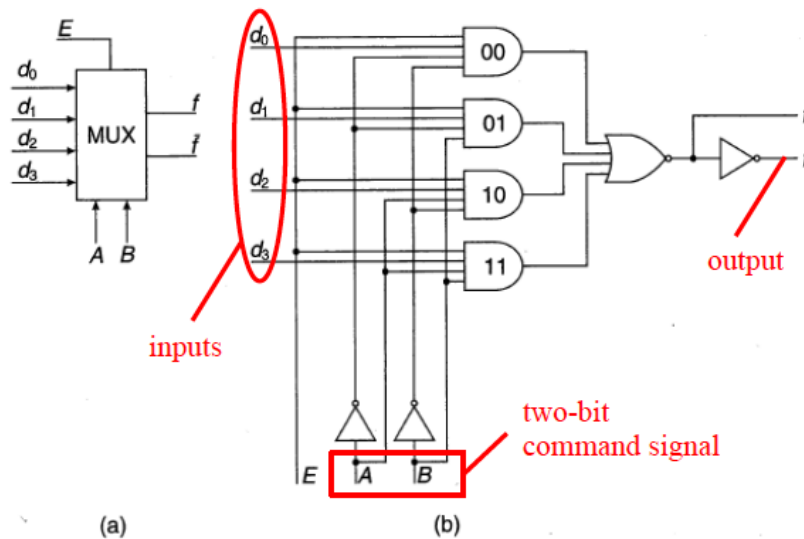


Figure 5.1 (a) Block diagram of a 4-input multiplexer and (b) its gate implementation

Holdsworth, Fig. 5.1 (annotated); Houh, ¶17.

¹⁵ *Id.*, [0005].

¹⁶ *See Id.*, [0006]-[0012].

¹⁷ Houh, ¶¶14-16; *see also* Ex. 1024, 5:28.

A demultiplexer is the opposite of a multiplexer—it selects one of a plurality of output lines for transmission of data received over a single input line. An example 4-output demultiplexer implementation is shown below:

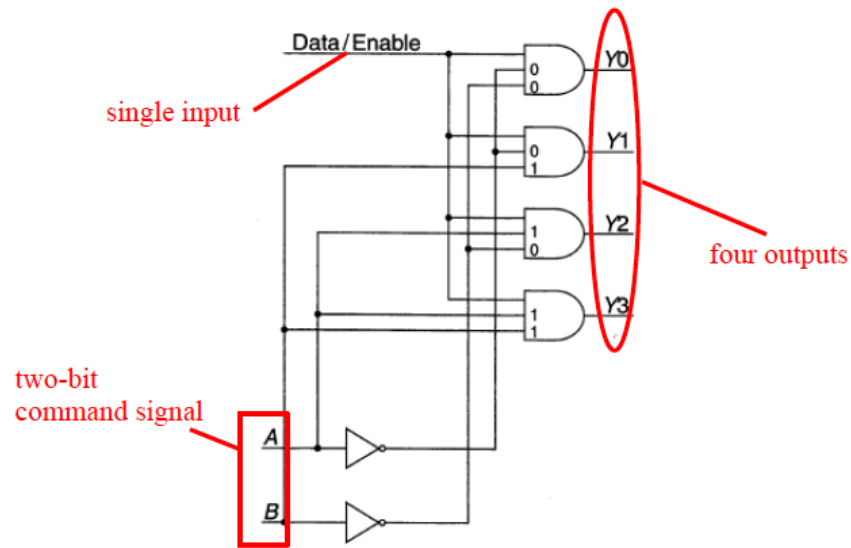


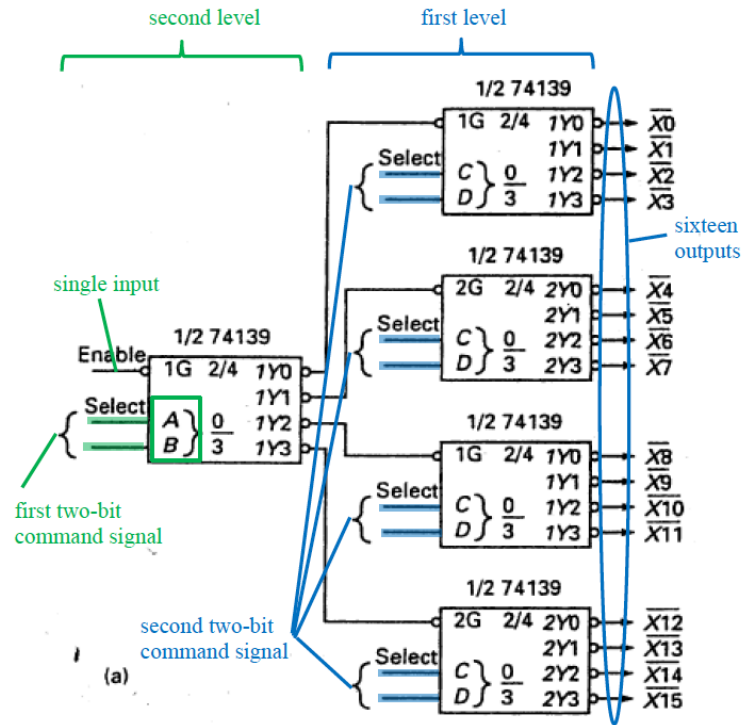
Figure 5.10 Basic demultiplexer

Holdsworth, Fig. 5.10 (annotated); Houh, ¶18.

Dr. Houh further explains the functionality of such multiplexers and demultiplexers.¹⁸

To form larger de-multiplexers with more output lines, it was known to use two-level structures comprised of several demultiplexers. For example, as shown below, it was known to use several 1:4 demultiplexers in a two-level structure to create a 1:16 demultiplexer.

¹⁸ Houh, ¶¶17-18.



Holdsworth, Fig. 5.17(a) (annotated); Houh, ¶19.

The first level (on the right) includes four 1:4 demultiplexers controlled by lines C and D. The second level (on the left) includes a single 1:4 demultiplexer controlled by lines A and B. Through this two-level structure, a four-bit value (A, B, C, D) can select any of the sixteen possible outputs (X0-X15) for a single input. It was also known to create large multiplexers using a similar two-level structure.¹⁹

Link aggregation uses demultiplexing to distribute data traffic from a single input among several available outputs. For example, a 1:16 demultiplexer such as the one shown above would distribute data traffic across 16 different parallel

¹⁹ See Ex. 1008, 3:61-7:19, Fig. 8A; Houh, ¶20.

output lines. A multiplexer then combines incoming data from all sixteen lines to a single line.²⁰

IV. SUMMARY OF THE '740 PATENT

A. Overview of the '740 Patent

The '740 Patent is “directed to communication networks, and particularly to methods and systems for link aggregation in network elements.”²¹ The patent describes a communication system 20 having a network element 32 that “interconnects a plurality of user ports 24 to a communication network 28.”²² “Network 28 may comprise a wide-area network (WAN), such as the Internet, a network internal to a particular organization (Intranet), or any other suitable communication network.”²³ The user ports may connect “to a user node, such as a layer 2 or layer 3 switch.”²⁴

The network element 32 includes “one or more user interface modules (UIMs), such as line cards 40. Each line card is assigned to process data frames of one or more user ports.”²⁵ Network element 32 is shown below in Fig. 2.

²⁰ Houh, ¶¶17-21.

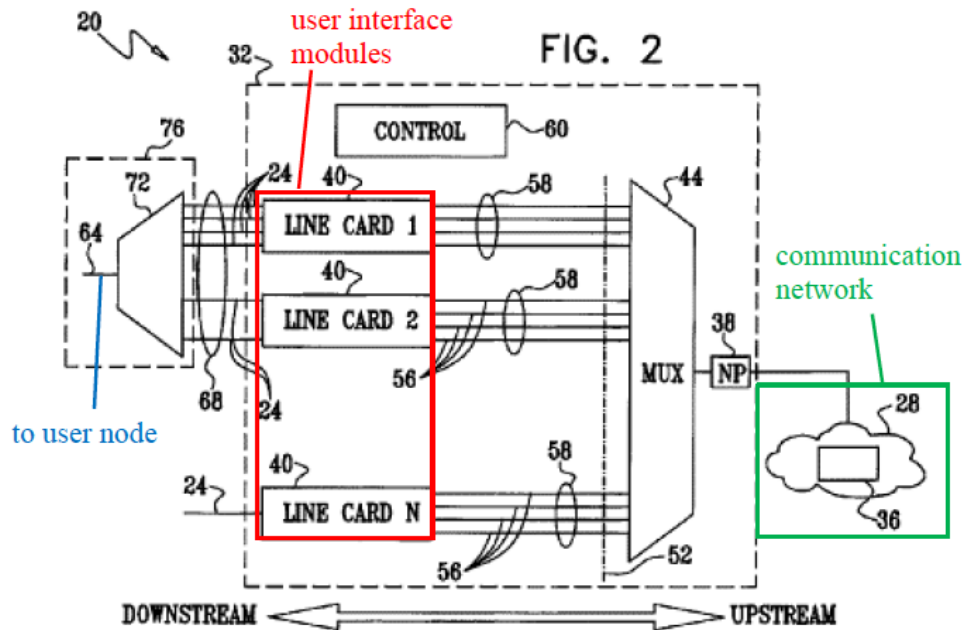
²¹ '740 Patent, 1:5-7.

²² *Id.*, 4:7-8.

²³ *Id.*, 4:9-11.

²⁴ *Id.*, 6:38-39.

²⁵ *Id.*, 4:28-30.



'740 Patent, Fig. 2 (annotated); Houh, ¶23.

The line cards 40 connect to a user node through a set of ports 24.²⁶ “User ports 24 forming port 64 are configured as an Ethernet LAG [link aggregation group] group, referred to as an external LAG group 68.”²⁷ The '740 Patent claims these ports 24 as “a first group of first physical links arranged in parallel.”²⁸

The interface modules connect to the communication network 28 using backplane traces 56: “Backplane 52 comprises physical links, such as backplane traces 56, typically in the form of printed circuit board (PCB) conductors.”²⁹ The

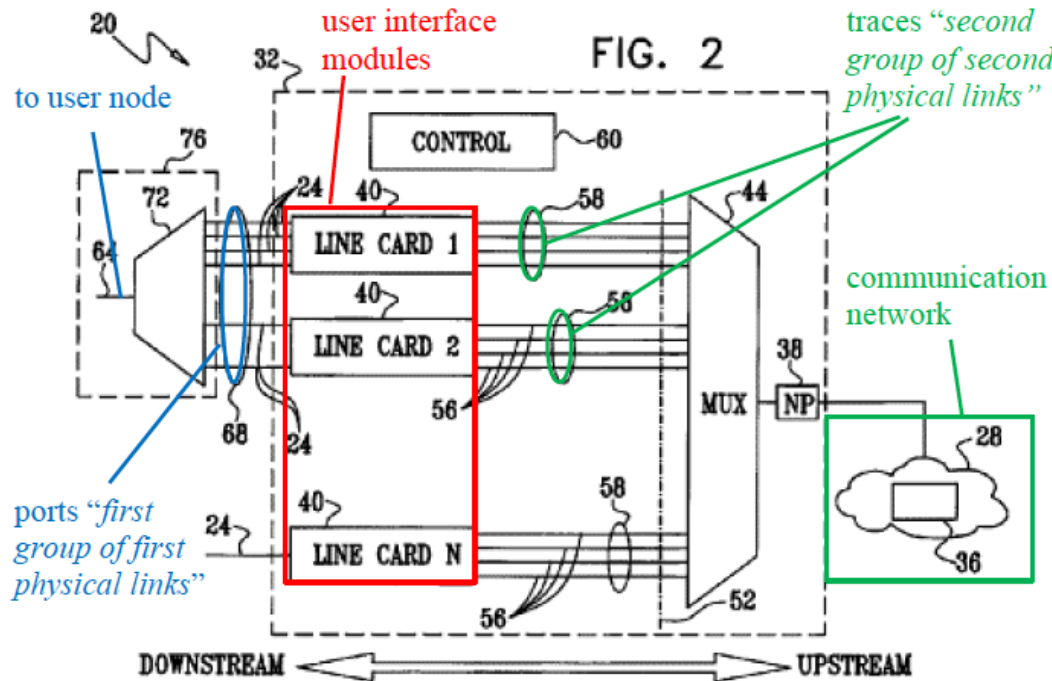
²⁶ See *id.*, 4:5-11, Fig. 2.

²⁷ *Id.*, 6:21-23.

²⁸ See, e.g., '740 Patent, claim 1.

²⁹ *Id.*, 4:34-37.

'740 Patent claims these traces 56 as “a second group of second physical links arranged in parallel.”³⁰



'740 Patent, Fig. 2 (annotated); Houh, ¶25.

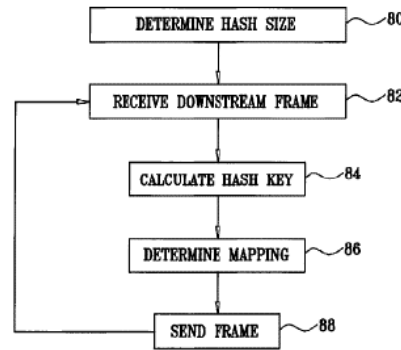
The network element 32 further includes a control module 60 that, for each received packet, determines which trace and which port will transmit that packet.³¹ Selection of both a port 24 from the first group and a trace 56 from the second group can be done in “single combined mapping operation that combines the two mapping operations described above.”³² The “combined mapping comprises a

³⁰ See *e.g.*, *id.*, claim 1.

³¹ See *id.*, 3:45-53.

³² *Id.*, 6:63-65.

single hashing operation that determines, for each such downstream frame, both the backplane trace 56 over which the frame is to be sent to one of line cards 40, and the user port 24 to be used within external LAG group 68.”³³ This combined mapping operation includes several steps, as shown below in Fig. 4.



'740 Patent, Fig. 4.

At step 80, “[t]he method begins with control module 60 determining a hashing size parameter.”³⁴ The hashing size parameter is based on the total number of traces multiplied by the total number of ports.³⁵ Then, at step 82, a data frame is received.³⁶ “For each downstream frame, control module 60 calculates a hashing key of the frame ... by applying a suitable hashing function to the frame attributes of the downstream frame.”³⁷ At step 86, “[m]odule 60 divides the hashing key of

³³ *Id.*, 6:66-7:3.

³⁴ *Id.*, 7:25-26.

³⁵ *See id.*, 7:27-30, 7:40-42.

³⁶ *Id.*, 7:43-45.

³⁷ *Id.*, 7:45-50.

the downstream frame by N_{bpow} [the hashing size parameter] and retains the modulo, or the remainder of the division operation, as a mapping index.”³⁸

Accordingly, the steps in Fig. 4 produce a single hash computation used to determine both a port and a trace to transmit the packet. The ’740 Patent provides an example in which “[m]odule 60 partitions the binary representation of the mapping index into two parts having N_1 and N_2 bits.”³⁹ The first set of bits N_1 is used to select one of the ports: “[m]odule 60 uses N_1 bits as a user slot/port index, indicating over which user port 24 in external LAG group 68 the frame should be sent.”⁴⁰ The second set of bits N_2 is used to select one of the traces: “[t]he remaining N_2 bits are used as a backplane trace index, indicating over which of the backplane traces of the relevant line card the frame should be sent.”⁴¹

As the below will show, a POSITA would’ve found the link aggregation concepts described and claimed in the ’740 Patent obvious over the prior art.⁴²

B. Prosecution History

The ’740 Patent was filed April 7, 2006. In a first non-final Office Action on October 2, 2008, the Office rejected many claims as anticipated by U.S. Patent

³⁸ *Id.*, 7:51-53.

³⁹ *Id.*, 7:60-61.

⁴⁰ *Id.*, 7:61-64.

⁴¹ *Id.*, 7:64-66.

⁴² Houh, ¶¶22-29.

No. 6,963,578.⁴³ Dependent claims 4, 5, 13, 15, 19, 20, 28, and 30 contained allowable subject matter.⁴⁴

In response, the Applicant rewrote the dependent claims with allowable subject matter in independent form.⁴⁵ The Applicant also amended some claims to include limitations requiring the recited communication links to be “bi-directional.”⁴⁶ The claims were then allowed with no statement of reasons for allowance.⁴⁷

C. Level of Ordinary Skill in the Art

A POSITA in April 2006 would’ve had a working knowledge of computer networking, including such commonly used technologies as Ethernet and the Internet protocol / transmission control protocol (“TCP/IP”) suite. A POSITA would’ve had a bachelor’s degree in electrical engineering, or an equivalent, and two years of professional experience in the field of communication networks, including link aggregation in communication networks. Lack of professional experience can be remedied by additional education, and vice versa.⁴⁸

⁴³ Ex. 1002, 83.

⁴⁴ *Id.*, 85.

⁴⁵ *Id.*, 76; *see also id.*, 59-69.

⁴⁶ *Id.*, 59-69.

⁴⁷ *Id.*, 38.

⁴⁸ Houh, ¶¶7-9.

D. Claim Construction

Claim terms in IPR are construed according to their “ordinary and customary meaning” to those of skill in the art.⁴⁹ For the purposes of this proceeding and the grounds presented herein, no claim term requires express construction.⁵⁰ For clarity, however, Petitioner notes below example embodiments in the specification of certain terms.⁵¹

1. “interface module”

Each independent claim of the ’740 Patent recites “one or more interface modules.” The patent explicitly contemplates that a line card is within the scope of the term interface module: “Network element 32 comprises one or more user interface modules (UIMs), such as line cards 40.”⁵² A line card is thus an interface module as claimed.⁵³

⁴⁹ 37 C.F.R. §42.100(b); *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc).

⁵⁰ *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017).

⁵¹ Houh, ¶¶30-32.

⁵² ’740 Patent, 4:28-29.

⁵³ Houh, ¶33.

2. “selecting, in a single computation”

As explained above at IV.A, the '740 Patent describes a multi-step selecting process as shown in Fig. 4 and explained in accompanying text. This multi-step process includes determining a hash size (step 80), calculating a hash key (step 84), and determining a mapping (step 86). This multi-step process is recited in various dependent claims—including claims 8-10 which recite that the selecting process includes multiple steps, including (1) “determining a hashing size responsively to a number of at least some of the first and second physical links,” (2) “applying the hashing function to the at least one of the frame attributes to produce a hashing key,” and (3) “calculating a modulo of a division operation of the hashing key by the hashing size.”

The “single computation” doesn’t refer to performing the selecting process in a single step, but rather to a single computational result of a hashing function (*e.g.*, single value) that selects the outcome.⁵⁴ Thus, the single value, or the final processing step that produces that single value, are examples of a “single computation.”⁵⁵

⁵⁴ See, *e.g.*, '740 Patent, 5:29-31.

⁵⁵ Houh, ¶¶34-36.

V. NO BASIS EXISTS FOR DISCRETIONARY DENIAL

Although institution is discretionary, the grounds for unpatentability presented in this Petition are neither cumulative or redundant to the prosecution of the '740 Patent. The prior art here wasn't previously considered and isn't redundant to any combination of references previously considered. Nor do the *Fintiv* or *General Plastic* factors provide any basis for denial.

A. Discretionary Denial Under *Fintiv* Isn't Appropriate

The six *Fintiv* factors considered for §314 denial favor institution.⁵⁶ Factor 1 is neutral. Patent Owner filed its complaint against Juniper asserting infringement of the '740 Patent on July 31, 2023. Because the district court proceeding is in its early stages, no motion for stay has been filed. Due to the early stage of the district court proceeding, the parties have only minimally invested in the district court proceeding, no case schedule currently exists, and no overlap in prior art issues exist. Indeed, Juniper has yet to respond to the complaint. A final written decision will issue from this IPR well before the 27-month median time to trial in Delaware elapses. Factors 2-4 therefore weight in favor of institution. The parties are the same so factor 5 doesn't provide a basis for denying institution. And this Petition seeks joinder of already instituted proceedings. The institution of the original

⁵⁶ See *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 (PTAB Mar. 20, 2020) (precedential).

proceeding, IPR2023-00401, confirms the compelling evidence of unpatentability included in this petition which relies on the same evidence. Thus, factor 6 also weighs in favor of institution. And *Fintiv* offers no basis for denying institution here.

B. Discretionary Denial Under *General Plastic* Isn't Appropriate

Although the Board applies *General Plastic* to petitions seeking joinder, those factors don't support denying institution here.⁵⁷ This Petition is Petitioner's first challenge to the '740 Patent. And as its Motion for Joinder details, Petitioner agrees to assume a passive understudy role so that its participation won't affect the schedule of IPR2023-00401 or impact the Board's finite resources. In doing so, Petitioner "effectively neutraliz[es] the *General Plastic* [discretionary denial] factors."⁵⁸ *General Plastics* therefore offers no basis for discretionary denial.⁵⁹

⁵⁷ *Cf. Apple Inc. v. Uniloc 2017 LLC*, IPR2020-00854, Paper 9 (P.T.A.B. Oct. 28, 2020) (precedential) (applying *General Plastic* factors where Petitioner previously filed unsuccessful IPR petitions).

⁵⁸ *Apple Inc. v. Uniloc 2017 LLC*, IPR2018-00580, Paper 13 at 10-11 (P.T.A.B. Aug. 21, 2018).

⁵⁹ *See General Plastic Industrial Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19 at 9-10 (P.T.A.B. Sept. 6, 2017) (precedential) (listing factors considered when petitioners have previously filed petitions).

VI. GROUNDS 1 AND 2: BRUCKMAN ALONE OR IN VIEW OF BASS RENDER OBVIOUS CLAIMS 1-31.

Ground 2 differs from Ground 1 by relying on additional obviousness teachings from Basso in limitation [1.6].⁶⁰ Both Grounds 1 and 2 confirm that the Challenged Claims are obvious over the prior art.

A. Bruckman Summary

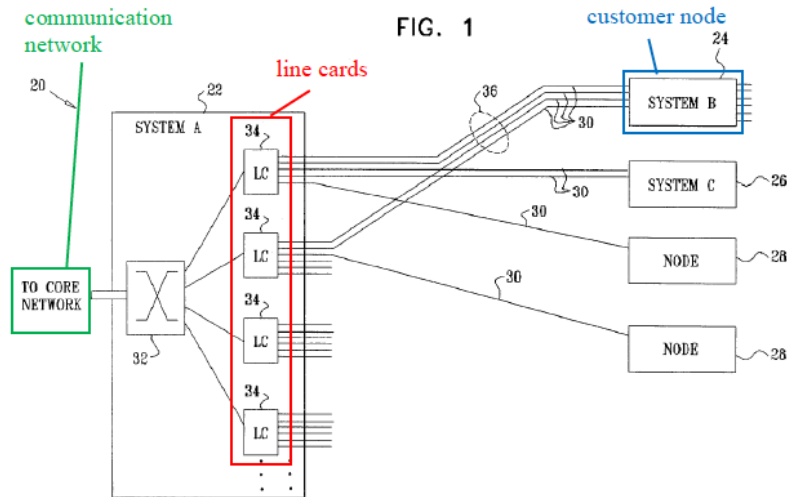
Bruckman relates “to data communication systems, and specifically to methods and systems for link aggregation in a data communication network.”⁶¹ Bruckman describes a communication system 20 having Equipment 22 that “is configured to convey packet data traffic between the customer nodes and a network (which may be a metro network, access network, or other type of core network, for example).”⁶² Equipment 22 further includes “a main switching card 32, which is connected to multiple line cards 34.”⁶³ As shown in Fig. 1 below, the main switching card 32 and line cards 34 form a two-level multiplexer/demultiplexer.

⁶⁰ Houh, ¶¶37-39.

⁶¹ Bruckman, [0001].

⁶² *Id.*, [0047].

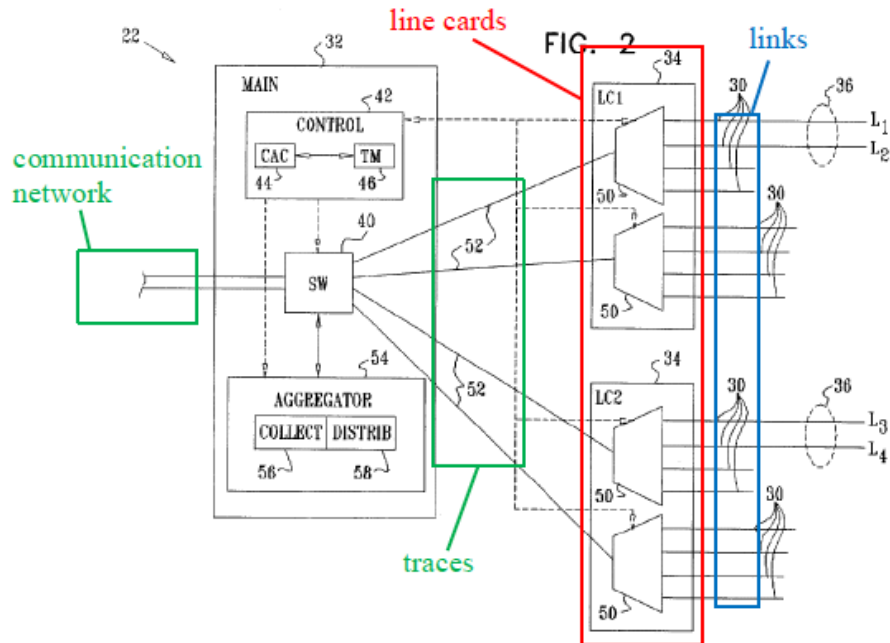
⁶³ *Id.*, [0047].



Bruckman, Fig. 1 (annotated); Houh, ¶40.

Fig. 2 (shown below) provides more details of the communication system 20 shown in Fig. 1. The line cards 34 in Fig. 2 connect to the customer node via links 30. “Each line card 34 comprises one or more concentrators 50, which comprise multiple ports that serve respective links 30.”⁶⁴ The line cards 34 connect to the switching core 40 and core network through backplane traces 52. Bruckman thus describes a two-level multiplexing/demultiplexing structure in which the first level includes the concentrators 50, each of which selects from one of four outputs (links 30) and the second level includes switching core 40, which selects one of a plurality of outputs (traces 52).

⁶⁴ *Id.*, [0056].



Bruckman, Fig. 2 (annotated); Houh, ¶41.

When processing traffic from the customer nodes to the communication network, Bruckman’s two-level structure acts as a multiplexer: “The concentrators multiplex data traffic between links 30 and traces 52, which connect the concentrators to switching core 40.”⁶⁵ When processing traffic from the communication network to a customer node, Bruckman’s two-level structure acts as a demultiplexer by determining which trace and which port will transmit that data frame. The main card 32 includes a distributor 58, that “determines the link over which to send each frame based on information in the frame header.”⁶⁶ To

⁶⁵ *Id.*, [0056].

⁶⁶ *Id.*, [0058].

make this selection, the distributor “applies a predetermined hash function to the header information.”⁶⁷ Bruckman provides an example of such a hash function shown in Table 1 below.

```
A unsigned short hash(unsigned char *hdr, short lagSize)
    {
      short i;
      unsigned short hash=178; // initialization value
      for (i=0; i<4; i++)      // hdr is 4 bytes length
B      hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
C      return (hash % lagSize)
    }
```

Bruckman, Table 1 (annotated); Houh, ¶42.

Like the selection process in the '740 Patent, Bruckman's hashing function includes multiple steps that produce a single hash computation. First, as shown in line A, the hashing function receives the lagSize parameter, which “is the number of active ports (available links 30) in link aggregation group.”⁶⁸ In line B, an operation is applied to header information (in variable *hdr*⁶⁹) to create a hashing key stored in the “hash” variable. Then, in line C, a modulo of the division of the

⁶⁷ *Id.*, [0058].

⁶⁸ *Id.*, [0064].

⁶⁹ *Id.*, [0064] (“Here *hdr* is the header of the frame to be distributed.”).

hash key and the LAG size is produced. This final, single modulo value is then used to select which link 30 will transmit the data frame.⁷⁰

Accordingly, Bruckman shows that using a set of parallel traces and a set of parallel links/ports to transmit packets was known before the filing of the '740 Patent. Bruckman additionally shows that it was known to apply a hashing function to select one of the links.⁷¹

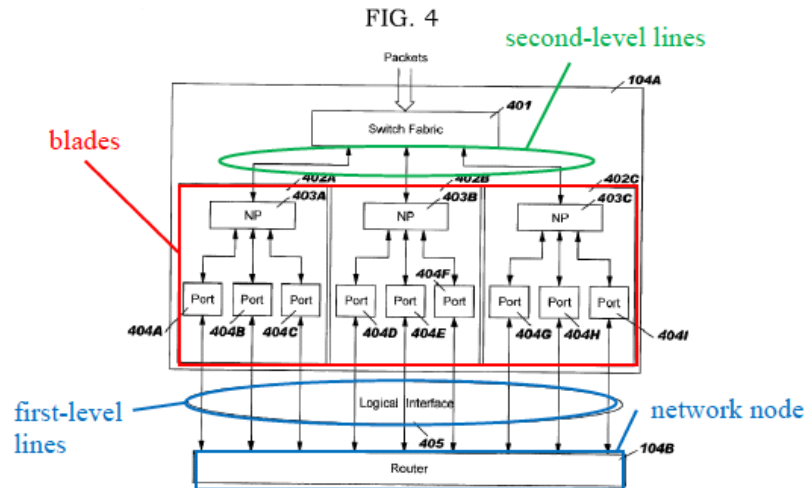
B. Basso Summary

Basso relates to link aggregation, particularly, “the field of packet switching networks, and more particularly to logically grouping physical ports of a network device.”⁷² Basso describes a network device, such as a router, that passes packets between the Internet and a client device. Like Bruckman, Basso describes a two-level structure, as shown in Fig. 1 below.

⁷⁰ *Id.*, [0058].

⁷¹ Houh, ¶¶40-44.

⁷² Basso, [0001].



Basso, Fig. 4 (annotated); Houh, ¶45.

Basso’s network device 104a includes “a switch fabric coupled to a plurality of blades where each blade may comprise one or more network processors coupled to one or more ports.”⁷³ As shown in Fig. 4, Basso’s two-level structure includes a first group of physical links—each associated with one of the ports 404—that connect the network device to a network node (in this case, Router 104b).

Additionally, a second set of links connects each of the blades to the switching fabric: “a switch fabric 401 configured to direct the incoming packets of data to particular blades 402A-C coupled to switch fabric 401.”⁷⁴

⁷³ *Id.*, [0009].

⁷⁴ *Id.*, [0030].

Upon receipt of a packet, “a hash function may be performed on the source and destination address in the packet header to generate a hash value.”⁷⁵ Using that single hash value, “an appropriate blade/port combination may be identified to transmit the received packet of data.”⁷⁶ Basso thus uses a single hash computation to select both a first-level line (one of the ports) and second-level line (one of the blades)—a blade/port combination.⁷⁷

C. Reasons to Combine Bruckman and Basso

A POSITA would’ve found it obvious that when using a two-level demultiplexer structure such as the one described by Bruckman, a single hash computation controls both a first-level and a second-level demultiplexer. Basso provides explicit evidence of doing so.⁷⁸

Both Bruckman and Basso are analogous art to the ’740 Patent. Bruckman and Basso are directed to the same field of endeavor as the ’740 Patent—network communications.⁷⁹

⁷⁵ *Id.*, [0011].

⁷⁶ *Id.*, [0012].

⁷⁷ Houh, ¶¶45-47.

⁷⁸ *Id.*, ¶48.

⁷⁹ *Compare* ’740 Patent, 1:5-7 (“The present invention relates generally to communication networks, and particularly to methods and systems for link aggregation in network elements”) *with* Bruckman, [0001] (“The present invention

Bruckman’s hashing technique produces a single hash computation that is used to determine which link 30 receives a data frame: the “distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36 ... determines the link over which to send each frame based on information in the frame header.”⁸⁰

Bruckman provides limited description of how the hash computation is used to select a trace. Basso provides further details regarding such two-level selections and includes explicit evidence of using a single hash computation to make a selection in both levels of a two-level multiplexer/demultiplexer structure. As explained above, Basso’s network device uses one hash value to select “an appropriate blade/port combination ... to transmit the received packet of data.”⁸¹ Basso thus uses a single hash computation to select a blade/port combination—both a first-level line (one of the ports) and second-level line (one of the blades). Accordingly, to the extent a POSITA implementing the device of Bruckman

relates generally to data communication systems, and specifically to methods and systems for link aggregation in a data communication network.”) *and* Basso, [0001] (“The present invention relates to the field of packet switching networks, and more particularly to logically grouping physical ports of a network device, e.g., router, into logical interfaces.”); *see also* Houh, ¶48.

⁸⁰ Bruckman, [0058].

⁸¹ Basso, [0012].

needed more information about how a single hash computation would be used to control both levels of a two-level multiplexer/demultiplexer structure, Basso such teachings.

A POSITA would've had a reasonable expectation of success because Basso describes a substantially similar two-level multiplexer/demultiplexer structure as Bruckman.⁸² Thus, using a single hash computation to make selections in both levels of the two-level multiplexer/demultiplexer structure would've worked as well in Bruckman. A POSITA would've further found the combination beneficial as using a single hash computation (rather than multiple) would've allowed for improved operational efficiency due to the simplified design.⁸³ Calculating a single hash value as opposed to multiple hash values reduces the computational load and thus allows for faster switching, which is consistent with Bruckman's goal to "ensure that sufficient bandwidth will be available on the links in the group in order to meet service guarantees."⁸⁴

Thus, the combination of Bruckman and Basso represents the application of a known technique (Basso's single hash computation to select both levels in a two-level multiplexer/demultiplexer structure) to Bruckman's method (using a single

⁸² Houh, ¶52.

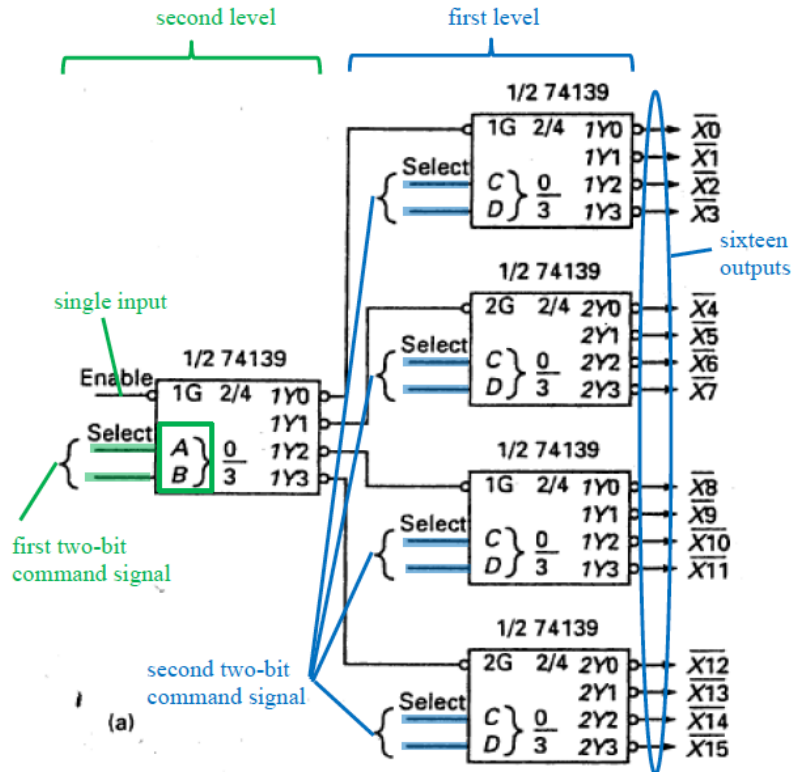
⁸³ *Id.*, ¶52.

⁸⁴ Bruckman, [0015].

hash computation for a two-level multiplexer/demultiplexer structure) to yield predictable results (fast and computationally efficient switching).⁸⁵

D. Holdsworth Summary

Holdsworth is a textbook entitled “Digital Logic Design.” Holdsworth describes a basic multiplexer and a basic demultiplexer.⁸⁶ Holdsworth further describes how 1:4 demultiplexers may be used in a two-level fashion to create a 1:16 multiplexer, as shown in Fig. 5.17(a) below.



Holdsworth, Fig. 5.17(a) (annotated); Houh, ¶54.

⁸⁵ Houh, ¶¶48-53.

⁸⁶ See Holdsworth, Figs. 5.1, 5.10.

Holdsworth thus shows that when using two levels of 1:4 demultiplexers to create a 1:16 demultiplexer, a designer would use a first two-bit signal for selection at the first level and a second two-bit signal for selection at the second level. In other words, Holdsworth shows that it was a well-known technique to use a first subset of bits for selection at the first level of a two-level demultiplexer, and a second subset of bits for selection at the second level of the two-level demultiplexer.

Holdsworth is analogous art to the '740 Patent, as its engineering principles are applicable to the same field of endeavor as the '740 Patent—communication networks. As explained above at III.B, link aggregation in communication networks utilizes multiplexers, which Holdsworth describes in detail.⁸⁷ Because Holdsworth provides implementation details for multiplexers and demultiplexers, Holdsworth is reasonably pertinent to the technical problem of load balancing and link aggregation to which the '740 Patent relates.⁸⁸

E. Holdsworth Shows Background Knowledge of a POSITA

The Holdsworth textbook is “intended to cover all the material that is needed in a typical undergraduate or master’s course on Digital Logic Systems, and also to

⁸⁷ See generally, *id.*.

⁸⁸ Houh, ¶¶54-56.

act as a reference text for graduates working in this field.”⁸⁹ Holdsworth represents the background knowledge of a POSITA because Holdsworth’s subject matter would’ve been part of the undergraduate education of a POSITA.⁹⁰ For example, digital logic systems was a commonly required course in undergraduate electrical engineering degree plans.⁹¹

F. Claim 1

Regarding element [1.0], Bruckman “relates generally to data communication systems, and specifically to methods and systems for link aggregation in a data communication network.”⁹² Thus, because Bruckman describes methods for link aggregation in a data communication network, Bruckman renders obvious a “method for communication” as claimed.⁹³

For element [1.1], Bruckman describes a communication system that includes a set of networked customer nodes 24, 26, 28 (any of which is a “network node”) that are connected to multiple line cards 34 (“one or more interface modules”) over parallel physical links 30 (“a first group of physical links arranged in parallel”). “In this example, central office equipment 22 communicates with

⁸⁹ Holdsworth, Preface.

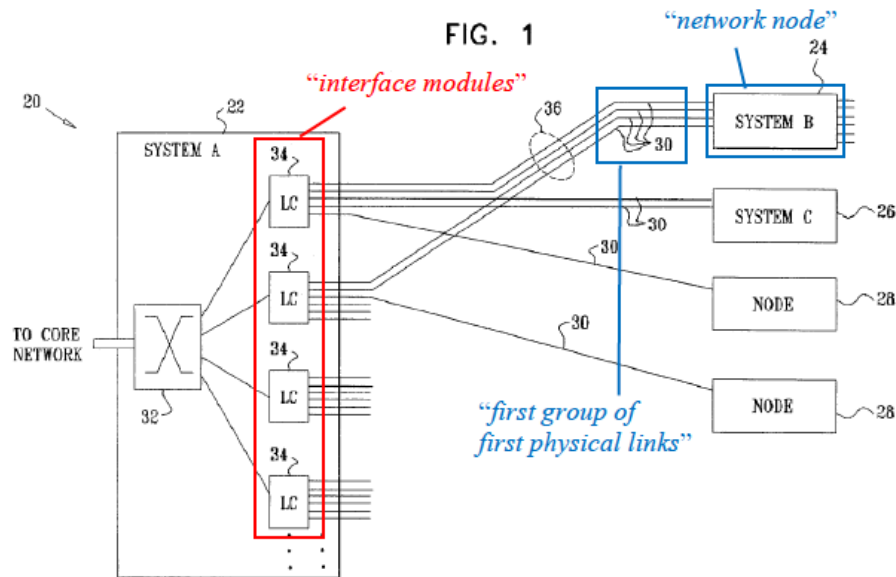
⁹⁰ Houh, ¶57.

⁹¹ See Ex. 1022.

⁹² Bruckman, Abstract, [0001].

⁹³ Houh, ¶¶58-59.

customer nodes 24, 26, 28 ... over physical links 30 ... For this purpose, equipment 22 comprises a main switching card 32, which is connected to multiple line cards 34 that serve links 30.”⁹⁴



Bruckman, Fig. 1 (annotated); Houh, ¶60.

The links 30 shown in Fig. 1 form an “aggregation group 36.”⁹⁵ “Link aggregation is a technique by which a group of parallel physical links between two endpoints in a data network can be joined together into a single logical link.”⁹⁶

Thus, because Bruckman describes connecting a customer node to a set of line cards over a set of parallel links, Bruckman renders obvious “coupling a

⁹⁴ Bruckman, [0047].

⁹⁵ *Id.*, [0057].

⁹⁶ *Id.*, [0002].

network node to one or more interface modules using a first group of first physical links arranged in parallel” as claimed.⁹⁷

Regarding element [1.2], Bruckman’s Equipment 22 transmits data over the physical links 30 in both directions. “Links 30 typically comprise full-duplex Ethernet links.”⁹⁸ The term full-duplex “[r]efers to a communication system or equipment capable of transmission simultaneously in two directions.”⁹⁹ Accordingly, links 30 transmit data in both directions.

For data traffic in the downstream direction (from communication network to customer node), Bruckman’s equipment 22 uses a distributor 58: “Aggregator 54 comprises a distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36.”¹⁰⁰ In the upstream direction (from customer node to communication network), Bruckman utilizes a collector 56: “Aggregator 54 further comprises a collector 56, which collects data frames that were received over different links 30 in group 36, and arranges the frames back into a single traffic stream.”¹⁰¹

⁹⁷ Houh, ¶¶60-62.

⁹⁸ Bruckman, [0047].

⁹⁹ Ex. 1009, 376; *see also* Ex. 1021, 8:11-13.

¹⁰⁰ Bruckman, [0058].

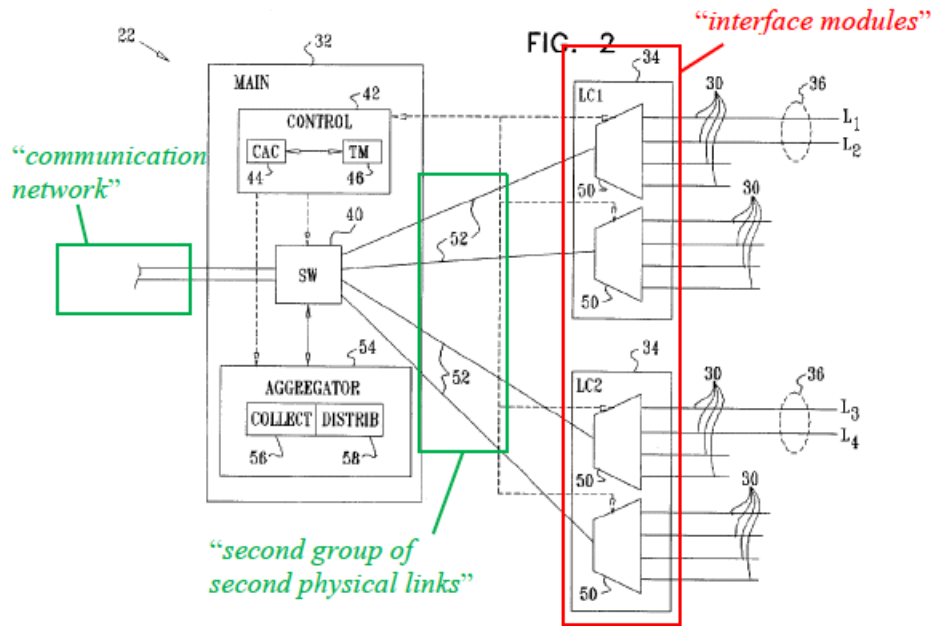
¹⁰¹ *Id.*, [0065].

Thus, because Bruckman describes traffic flowing in both directions across full-duplex links, Bruckman renders obvious “at least one of said first physical links being a bi-directional link operative to communicate in both an upstream direction and a downstream direction” as claimed.¹⁰²

For element [1.3], in Bruckman’s system, the line cards 34 (“one or more interface modules”) are coupled to the core network (“communication network”) with a set of parallel traces 52 (“a second group of second physical links arranged in parallel”). As shown in Fig. 2 below, “[e]ach line card 34 comprises one or more concentrators 50, which comprise multiple ports that serve respective links 30. The concentrators multiplex data traffic between links 30 and traces 52, which connect the concentrators to switching core 40.”¹⁰³

¹⁰² Houh, ¶¶63-66.

¹⁰³ Bruckman, [0056].



Bruckman, Fig. 2 (annotated); Houh, ¶67.

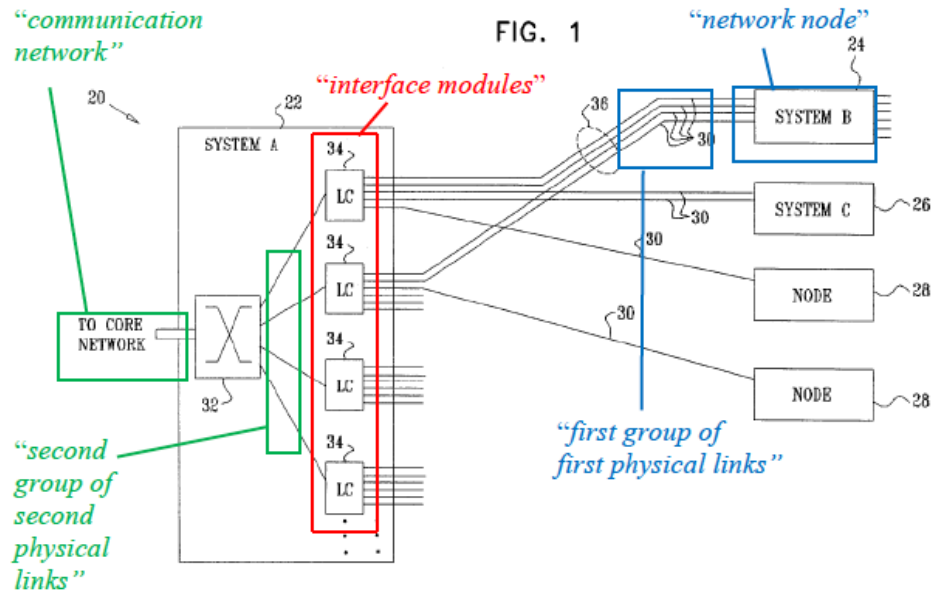
Each of the traces 52 connects the switching core 40 to one of the concentrators 50 and are arranged in parallel in Fig. 2. Traces 52 are part of a link aggregation group because Bruckman describes how “aggregation group extends over a number of concentrators 50,” and the traces 52 couple the concentrators to the switching core 40.¹⁰⁴ Since the traces 52 are part of a link aggregation group, Bruckman further renders obvious that the traces are “a group of parallel physical links.”¹⁰⁵

As shown in Fig. 1 below, the main card 32 connects to the core network. Thus, by connecting the line cards 34 (“interface modules”) to the switching core

¹⁰⁴ *Id.*, [0066]; *see also id.*, [0057].

¹⁰⁵ *Id.*, [0002].

40 of the main card 32, the traces 52 connect the line cards 34 to a “communication network” as claimed.



Bruckman, Fig. 1 (annotated); Houh, ¶69.

Thus, because Bruckman’s line cards connect to a core network using a set of parallel traces, Bruckman renders obvious “coupling each of the one or more interface modules to a communication network using a second group of second physical links arranged in parallel” as claimed.¹⁰⁶

For element [1.4], as discussed for [1.2], Bruckman’s system handles traffic in both directions between the core network and the customer nodes. An aggregator distributes traffic not only to full-duplex links 30 (see [1.2]), “but also

¹⁰⁶ Houh, ¶¶67-70.

to traces 52 that connect to multiplexers 50 that serve these links.”¹⁰⁷ Since the aggregator includes both a traffic “distributor” and a traffic “collector,” it would’ve been obvious that traffic flows in both directions over traces 52.¹⁰⁸

Accordingly, at least one of the traces 52 (“at least one of said second physical links”) is “a bi-directional link operative to communicate in both an upstream direction and a downstream direction” as claimed.¹⁰⁹

For element [1.5], Bruckman’s main card 32 includes an aggregator 54 that receives data frames that are sent between the core network (“communication network”) and links 30 connected to customer nodes (a “network node”). The data frames include various attributes, including frame headers (“frame attributes”): “Typically, distributor 58 determines the link over which to send each frame based on information in the frame header,” as described in the Background of the Invention.”¹¹⁰

Thus, because Bruckman’s main card 32 receives data frames from the core network for transmission to the links 30 which connect to customer nodes, and those data frames have headers, Bruckman renders obvious “receiving a data frame

¹⁰⁷ Bruckman, [0057].

¹⁰⁸ *Id.*, [0058], [0065]; Houh, ¶71.

¹⁰⁹ Houh, ¶72.

¹¹⁰ Bruckman, [0058].

having frame attributes sent between the communication network and the network node” as claimed.¹¹¹

Ground 1: Regarding element [1.6], first, Bruckman’s aggregator 54 includes a distributor 58 that “determines the link over which to send each frame.”¹¹² “Preferably, distributor 58 applies a predetermined hash function to the header information [‘frame attributes’].”¹¹³

Bruckman gives an example of a hash function to select a link in Table 1:

```

unsigned short hash(unsigned char *hdr, short lagSize)
{
    short i;
    unsigned short hash=178; // initialization value
    for (i=0; i<4; i++)      // hdr is 4 bytes length
        hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
    return (hash % lagSize);
}

```

“single computation”

Bruckman, Table 1, [0063] (annotated); Houh, ¶76.

Bruckman’s hashing function thus selects which link 30 (“first physical link out of the first group”) will transmit the data frame from the core network to the customer node based on the value calculated from the “hash % lagSize” operation (“in a single computation”). A POSITA would’ve understood that the final line of

¹¹¹ Houh, ¶¶73-74.

¹¹² Bruckman, [0058].

¹¹³ *Id.*

code in this hashing function returns the value of the operation “hash % lagSize.”¹¹⁴ A POSITA would’ve recognized that the code snippet provided in Table 1 is written using the syntax of the C or C++ programming languages. In both C and C++ languages, the “%” operator refers to a mathematical modulus operation, which returns the remainder from integer division.¹¹⁵ Either the modulus operation “hash % lagSize” itself or its result render obvious the claimed “single computation.”

Second, a POSITA would’ve found it obvious that Bruckman’s hashing function selects both which link 30 (“first physical link out of the first group”) and which trace 52 (“second physical link out of the second group”) will transmit the data frame from the core network to the customer node.¹¹⁶ Bruckman explicitly states that the hash function determines which link 30 receives the data frame: “distributor 58, which is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36 ... determines the link over which

¹¹⁴ Houh, ¶77.

¹¹⁵ See Ex. 1011, 85; Ex. 1020, 7:44-45; Houh, ¶77. Ex. 1011 has a December 18, 1992 publication date. See Ex. 1023.

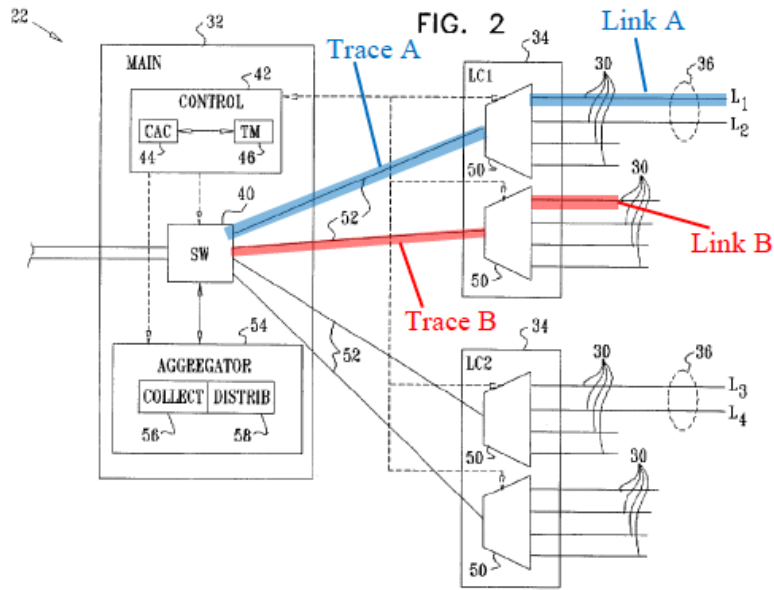
¹¹⁶ Houh, ¶78.

to send each frame based on information in the frame header ... Preferably, distributor 58 applies a predetermined hash function to the header information.”¹¹⁷

Bruckman provides limited description of how the hash computation is used to select a trace. A POSITA would’ve recognized, however, that Bruckman’s selection of a link 30 also determines which trace 52 will transmit the data frame.¹¹⁸ For example, referring to Fig. 2 below, if Link A is selected by the hash function, then Trace A is also selected since Trace A is the only trace that connects to link A to the switching core. Similarly, if Link B is selected by the hash function, then Trace B is also selected, since Trace B is the only trace that connects link B to the switching core. Accordingly, selection of a specific link 30 also determines a specific trace 52, e.g., whether to select Trace A or Trace B.

¹¹⁷ Bruckman, [0058].

¹¹⁸ Houh, ¶79.

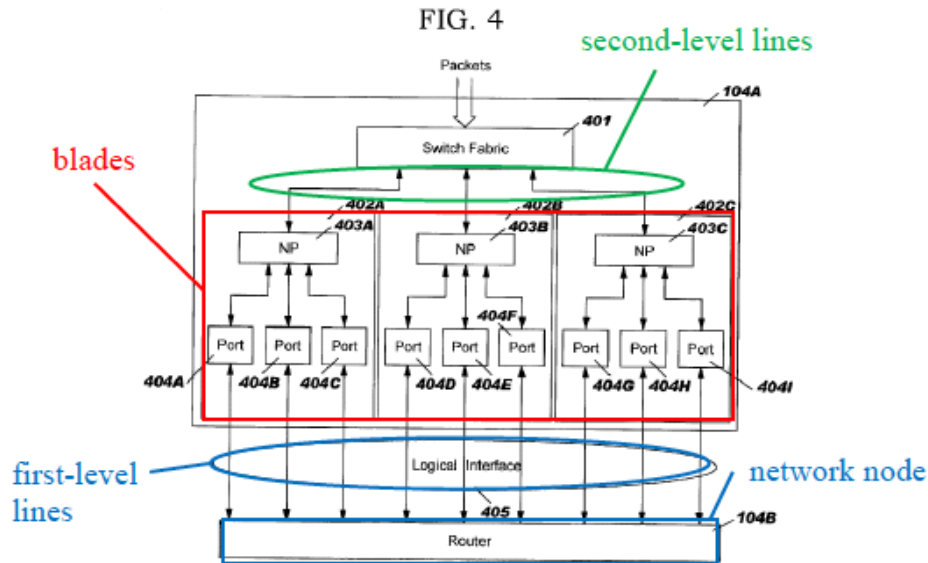


Bruckman, Fig. 2 (annotated); Houh, ¶79.

A POSITA would've found it obvious that selection of a link 30 also determines which trace 52 will transmit the data frame because each link 30 is associated with only a single trace 52.¹¹⁹

Ground 2: Alternatively, regarding element [1.6], it would've been obvious to use the result of a single hash computation to select links from two groups from the combination of Bruckman and Basso. Basso describes a two-level demultiplexer analogous to Bruckman's two-level demultiplexer, as shown in Fig. 4 below.

¹¹⁹ *Id.*, ¶80.



Basso, Fig. 4 (annotated); Houh, ¶82.

Like Bruckman’s links 30, Basso describes a set of physical links between blades of a router 104a to another network node (router 104B in the example of Fig. 4). Like Bruckman’s traces 52, each of Basso’s blades is connected to the switch fabric, which is connected to a communication network such as the Internet.¹²⁰

Basso describes using a single hash computation to select a blade/port combination. Upon receipt of a packet, “a hash function may be performed on the source and destination address in the packet header to generate a hash value.”¹²¹ Using that single hash value, Basso’s network device selects “an appropriate

¹²⁰ See Basso, Fig. 1.

¹²¹ *Id.*, [0011].

blade/port combination ... identified to transmit the received packet of data.”¹²²

Basso thus uses a single hash computation to select both a first-level line (one of the ports) and second-level line (one of the blades)—a blade/port combination.

Selection of Bruckman’s blade/port combination is analogous to selection of a trace 30 and link 52 in Bruckman. For the reasons explained above at VI.C, a POSITA would’ve found it obvious for Bruckman’s hash computation to select a trace and link combination, as evidenced by Basso’s hash computation that similarly selects a blade/port combination.

Thus, Bruckman alone or together with Basso renders obvious “selecting, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group” as claimed.¹²³

For element [1.7], Bruckman explains that its distributor “is responsible for distributing data frames arriving from the network among links 30 in aggregation group 36.”¹²⁴

¹²² *Id.*, [0012].

¹²³ Houh, ¶¶75-84.

¹²⁴ Bruckman, [0058].

Consistent with the example above in [1.6], it would've been obvious that sending a frame over a given link (such as Link B) involves sending the data frame over the single trace (such as Trace B) that leads to that link as well.¹²⁵

Thus, because Bruckman describes transmitting frames over a specific link based on the result of the hash function, Bruckman renders obvious “sending the data frame over the selected first and second physical links” as claimed.¹²⁶

Regarding element [1.8], as explained above for [1.2], [1.4], Bruckman's device transmits data frames in both directions, and thus it would've been obvious for links 30 and traces 52 to be bi-directional. A POSITA would've thus understood that traffic passed from the Bruckman's core network to the customer node is communicated across bi-directional links. Thus, Bruckman renders obvious “said sending comprising communicating along at least one of said bi-directional links” as claimed.¹²⁷

G. Claim 2

Regarding element [2.1], Bruckman's customer nodes 24, 26, and 28 are “user node[s]” as claimed. Indeed, Bruckman uses the term customer and user

¹²⁵ See [1.6]; Houh, ¶86.

¹²⁶ Houh, ¶¶85-87.

¹²⁷ *Id.*, ¶88.

interchangeably.¹²⁸ Thus, because Bruckman’s nodes 24, 26, and 28 are referred to as customer nodes, and Bruckman uses customer and user interchangeably, Bruckman renders obvious “wherein the network node comprises a user node” as claimed.¹²⁹

For element [2.2], first, as explained above at [1.7], Bruckman’s device transmits data frames to customer nodes over the links selected from the link aggregation group. Second, Bruckman’s disclosure is directed to “establishing a connection with a guaranteed bandwidth for transmitting data over a logical link that includes a plurality of parallel physical links between first and second endpoints.”¹³⁰ Bruckman explains that establishment of the aggregated links “provides the total bandwidth guaranteed by the customer’s service level agreement” and provides “for bandwidth allocation in a link aggregation system to

¹²⁸ Compare Bruckman, [0013] (“Service level agreements between network service providers and customers commonly specify a certain committed bandwidth”) with Bruckman, [0072] (“The connections on links 30, including any link aggregation groups, then compete for the remaining available bandwidth (typically in a weighted manner, based on the amount of excess bandwidth contracted for in the users’ service level agreements, as is known in the art)”).

¹²⁹ Houh, ¶¶89-90.

¹³⁰ Bruckman, Abstract; see also *id.*, claims 1, 11, 19, 32.

ensure that sufficient bandwidth will be available on the links in the group in order to meet service guarantees.”¹³¹

A POSITA would’ve understood that the presence of an agreement means that there would’ve been a service established before the frames are actually sent and that sending the frames is in furtherance of that established service.¹³²

Alternatively, a POSITA would’ve found it obvious for the data traffic sent via Bruckman’s system to include packets used to establish a common network communication service, a transmission control protocol (TCP) session.¹³³ As of the priority date, TCP communications were one of the most common type of network communication service in use.¹³⁴

Thus, Bruckman’s communication of data frames across the aggregated links provides a service between the communication network and customer nodes, which renders obvious “wherein sending the data frame comprises establishing a communication service between the user node and the communication network” as claimed.¹³⁵

¹³¹ *Id.*, [0014]-[0015].

¹³² Houh, ¶93.

¹³³ *Id.*, ¶93 (citing Ex. 1018).

¹³⁴ Houh, ¶93.

¹³⁵ *Id.*, ¶¶91-94.

H. Claim 3

For element [3.1], first, Bruckman’s traces 52 (“second physical links”) are part of a “printed circuit back plane” and are thus “backplane traces formed on a backplane” as claimed. “Typically, main card 32 and line cards 34 are arranged in a card rack and plug into a printed circuit back plane, (not shown) which comprises traces 52.”¹³⁶ Second, because the line cards 34 (“one or more interface modules”) are “arranged in a card rack and plug into [the] printed circuit back plane,” the line cards are “coupled” to the printed circuit back plane.¹³⁷

Thus, Bruckman renders obvious “wherein the second physical links comprise backplane traces formed on a backplane to which the one or more interface modules are coupled” as claimed.¹³⁸

I. Claim 4

For the same reasons discussed above, elements [4.0]-[4.5] are obvious over the prior art.¹³⁹

Regarding element [4.6], first, Bruckman explains that the links 30 (“first group of physical links”) may be organized into link aggregation groups. “For

¹³⁶ Bruckman, [0056].

¹³⁷ *Id.*

¹³⁸ Houh, ¶¶95-97.

¹³⁹ *See* [1.0]-[1.7]; Houh, ¶¶98-103.

example, an aggregation 50 group 36 of four physical links is defined between equipment 22 and node 24.”¹⁴⁰ “In the example shown in FIG. 2, aggregation group 36 comprises links L1 and L2, which are connected to LC1, and links L3 and L4, which are connected to LC2.”¹⁴¹ Second, Bruckman explains that “the embodiments described herein refer specifically to link aggregation in Ethernet (IEEE 802.3) networks.”¹⁴²

Thus, because Bruckman’s set of links 30 are Ethernet links that form a link aggregation group, Bruckman renders obvious “at least one of the first and second groups of physical links comprising an Ethernet link aggregation (LAG) group” as claimed.¹⁴³

J. Claim 5

For the same reasons discussed above, elements [5.0]-[5.5] are obvious over the prior art.¹⁴⁴ Similarly, the prior art discloses element [5.6].¹⁴⁵

¹⁴⁰ Bruckman, [0048].

¹⁴¹ *Id.*, [0057].

¹⁴² *Id.*, [0017]; *see also id.*, [0003] (“For Ethernet networks, link aggregation is defined by Clause 43 of IEEE Standard 802.3”).

¹⁴³ Houh, ¶¶104-106.

¹⁴⁴ *See* [1.0]-[1.7]; Houh, ¶¶107-112.

¹⁴⁵ *See* [4.6]; Houh, ¶113.

Regarding element [5.7], Bruckman explains that “[l]ink aggregation offers benefits of increased bandwidth.”¹⁴⁶ Thus, because link aggregation is used to increase bandwidth, Bruckman renders obvious using a LAG “so as to increase a data bandwidth provided to the network node” as claimed.¹⁴⁷

K. Claim 6

For element [6.1], Bruckman’s line cards 34 (“interface modules”) include concentrators that “multiplex data traffic between links 30 and traces 52.”¹⁴⁸ Bruckman also uses the term concentrators and multiplexers interchangeably.¹⁴⁹

Thus, because Bruckman’s concentrators/multiplexers 50 multiplex traffic from the customer nodes to the communication network, Bruckman renders obvious “wherein coupling each of the one or more interface modules to the communication network comprises at least one of multiplexing upstream data frames sent from the network node to the communication network” as claimed.¹⁵⁰

¹⁴⁶ Bruckman, [0002].

¹⁴⁷ Houh, ¶¶114-115.

¹⁴⁸ Bruckman, [0056].

¹⁴⁹ *See id.*, [0056] (referring to “concentrators 50”), [0057] (referring to “multiplexers 50”).

¹⁵⁰ Houh, ¶¶116-118.

Regarding element [6.2], a POSITA would've understood that demultiplexing is the opposite of multiplexing.¹⁵¹ As explained above at [1.2] and [1.4], Bruckman's links and traces are bi-directional.¹⁵² A POSITA would've understood that when the line card 34 receives a signal on a particular trace, and then selects one of several links with which to transmit the signal, the line card 34 is performing a demultiplexing function.¹⁵³ In other words, while traffic in the upstream direction (customer node to communication network) is multiplexed by the line cards 34, downstream traffic (communication network to customer node) is demultiplexed by the line cards 34.

Thus, because Bruckman's links are bi-directional and traffic in the downstream direction is demultiplexed by the line cards, Bruckman renders obvious "demultiplexing downstream data frames sent from the communication network to the network node" as claimed.¹⁵⁴

L. Claim 7

For element [7.1], Bruckman's hashing technique for selecting a link "distributes traffic in an approximately uniform manner across the entire set of

¹⁵¹ See *id.*, ¶119.

¹⁵² See Bruckman, [0058].

¹⁵³ Houh, ¶119.

¹⁵⁴ *Id.*, ¶120.

possible hash values.”¹⁵⁵ By distributing the load equally among possible hash values (which correspond to specific links), Bruckman’s hashing technique is designed to balance data among the links. Moreover, Bruckman incorporates by reference known algorithms for load balancing data traffic among links.¹⁵⁶

Thus, because Bruckman’s hashing technique seeks to distribute the load across the links in a uniform manner, Bruckman renders obvious “wherein selecting the first and second physical links comprises balancing a frame data rate among at least some of the first and second physical links” as claimed.¹⁵⁷

M. Claim 8

Regarding element [8.1], as explained above at [1.6], Bruckman’s selection process includes applying a hashing function to the header. Bruckman’s hashing function is shown in Table 1 below:

¹⁵⁵ Bruckman, [0061].

¹⁵⁶ *See id.*, [0005].

¹⁵⁷ Houh, ¶¶121-122.

hashing
function
“mapping
function” →

DISTRIBUTOR HASH FUNCTION

```

unsigned short hash(unsigned char *hdr, short lagSize)
{
    short i;
    unsigned short hash=178; // initialization value
    for (i=0; i<4; i++)      // hdr is 4 bytes length
        hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
    return (hash % lagSize)
}

```

Bruckman, Table 1 (annotated); Houh, ¶123.

Bruckman’s hashing function is a mapping function because it maps a particular data frame to a specific trace 52 and link 30 through the system. Further, claim 9 explicitly recites that a hashing function is within the scope of a “mapping function.”

Thus, because Bruckman’s selecting process involves a hashing function, Bruckman renders obvious “wherein selecting the first and second physical links comprises applying a mapping function to the at least one of the frame attributes” as claimed.¹⁵⁸

N. Claim 9

For element [9.1], as explained above at [8.1], Bruckman’s hash function is a mapping function. Thus, Bruckman renders obvious “wherein applying the mapping function comprises applying a hashing function” as claimed.¹⁵⁹

¹⁵⁸ *Id.*, ¶¶123-125.

¹⁵⁹ *Id.*, ¶¶126-127.

O. Claim 10

For element [10.1], as shown in line A, Bruckman’s hash function receives two parameters: *hdr* and *lagSize*. Bruckman explains that “*lagsize* is the number of active ports (available links 30) in link aggregation group.”¹⁶⁰

DISTRIBUTOR HASH FUNCTION

A unsigned short hash(unsigned char *hdr, short lagSize) “determining a hashing size”

```

{
    short i;
    unsigned short hash=178; // initialization value
    for (i=0; i<4; i++)      // hdr is 4 bytes length
        hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
    return (hash % lagSize)
}
```

Bruckman, Table 1 (annotated); Houh, ¶128.

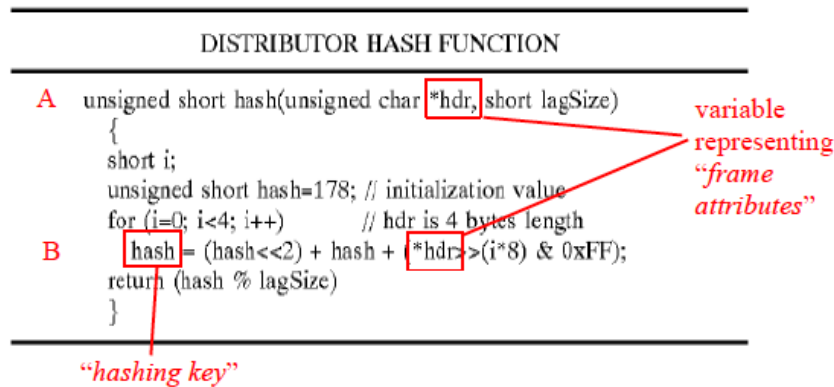
Accordingly, by receiving the *lagSize* parameter, which is the number of active ports (available links 30) in the link aggregation group, Bruckman’s hashing technique determines “a hashing size responsively to a number of at least some of the first and second physical links.” Thus, Bruckman renders this limitation obvious.¹⁶¹

For element [10.2], as shown on line A of the figure below, Bruckman’s hashing function receives the “*hdr*” parameter (“frame attributes”). “Here *hdr* is the

¹⁶⁰ Bruckman, [0064].

¹⁶¹ Houh, ¶129.

header of the frame to be distributed.”¹⁶² Then, in Line B, the *hdr* value is used to produce a “hashing key” represented by the variable “hash.”



Bruckman, Table 1 (annotated); Houh, ¶130.

The “hash” variable represents a hashing key because it is used in a modulus operation (%) to produce the final hash result.¹⁶³ A POSITA would’ve been familiar with programming terminology and recognized that in many programming languages, including the code shown in Table 1, the % symbol represents the modulus operation.¹⁶⁴

Thus, because Bruckman’s hashing function uses the header information to produce a hashing key, Bruckman renders obvious “applying the hashing function to the at least one of the frame attributes to produce a hashing key” as claimed.¹⁶⁵

¹⁶² Bruckman, [0064].

¹⁶³ See [1.6], [10.3].

¹⁶⁴ See Ex. 1011, 85; Houh, ¶131.

¹⁶⁵ Houh, ¶132.

For element [10.3], as shown in the annotated Bruckman hashing algorithm below, Line C includes the operation “hash % lagSize.” As known to POSITAs, the % operator represents the modulus (remainder) operation used to calculate the remainder from integer division.¹⁶⁶ A POSITA would’ve thus understood that the operation in Line C calculates the remainder (“modulo”) that results from dividing the value of “hash” (“hashing key”) by the value of “lagSize” (“hashing size”).¹⁶⁷

DISTRIBUTOR HASH FUNCTION	
A	unsigned short hash(unsigned char *hdr, short lagSize)
	{
	short i;
	unsigned short hash=178; // initialization value
	for (i=0; i<4; i++) // hdr is 4 bytes length
B	hash = (hash<<2) + hash + (*hdr>>(i*8) & 0xFF);
C	return (hash % lagSize);
	}

“hashing key” (pointing to hash) “hashing size” (pointing to lagSize)

Bruckman, Table 1.

Bruckman’s hashing function thus calculates and returns a “modulo of a division operation.” Where there are 16 different links (as in Bruckman’s example), the value of the lagSize parameter (“hashing size”) would be 16.¹⁶⁸

¹⁶⁶ See Ex. 1011, 85; Ex. 1020, 7:44-45; *supra* [1.6].

¹⁶⁷ Houh, ¶133.

¹⁶⁸ *Id.*, ¶134.

Thus, Bruckman’s hashing function, which returns “hash % lagSize,” renders obvious “calculating a modulo of a division operation of the hashing key by the hashing size” as claimed.¹⁶⁹

Regarding element [10.4], for the reasons explained above at [1.6], Bruckman’s application of a hashing function teaches selecting first and second physical links responsive to the single hash result. Bruckman’s hashing function returns a modulo, as shown in [10.3] by the line “return (hash % lagSize).” The modulus operation, or its resulting value (the modulo), is the single computation that is used to select a trace and a link as described above at [1.6]. Thus, Bruckman renders obvious “selecting the first and second physical links responsively to the modulo” as claimed.¹⁷⁰

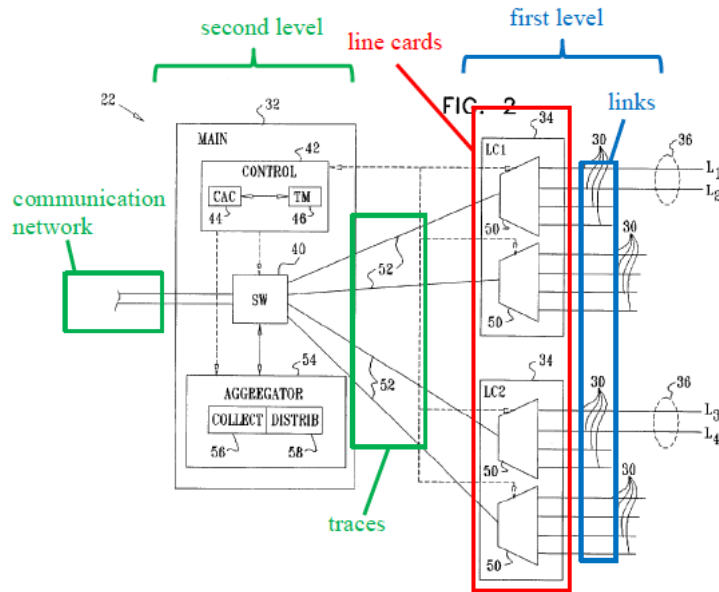
P. Claim 11

Regarding element [11.1], this claim limitation recites an obvious aspect of digital logic design. As explained above at VI.A, Bruckman’s equipment 22 is a two-level structure that resembles a two-level demultiplexer. Bruckman’s concentrators 50 act as first-level demultiplexers as each concentrator 50 selects one of four links 30 to transmit data received from a single trace 52. Bruckman’s

¹⁶⁹ *Id.*, ¶135.

¹⁷⁰ *Id.*, ¶136.

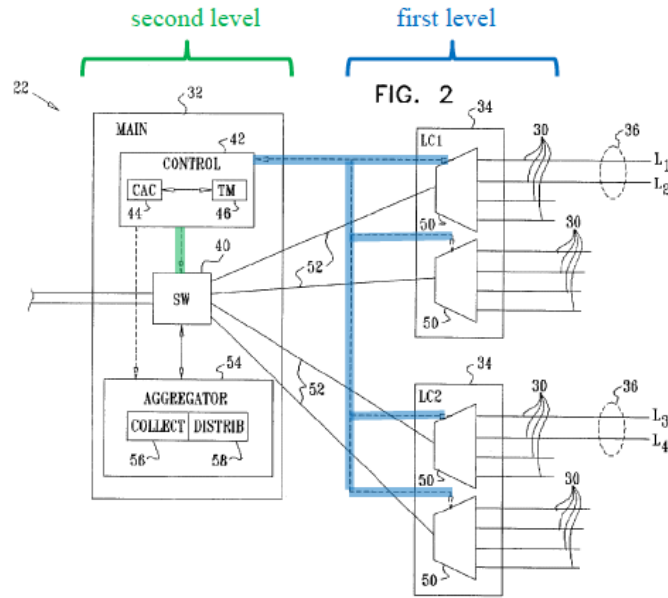
switching core 40 acts as the second-level demultiplexer by distributing data frames from the input channel amongst four separate traces 52.



Bruckman, Fig. 2 (annotated); Houh, ¶137.

Given this well-known, two-level structure, a POSITA would've found it obvious for Bruckman's Equipment 22 to be controlled in a manner consistent with well-known two-level demultiplexer control techniques.¹⁷¹ Indeed, Bruckman's equipment 22 includes a control line from the controller 42 to the switching core (second-level demultiplexer) shown below in green. Equipment 22 also includes control lines from the controller 42 to each of the concentrators (first-level demultiplexers) shown below in blue.

¹⁷¹ *Id.*, ¶138.



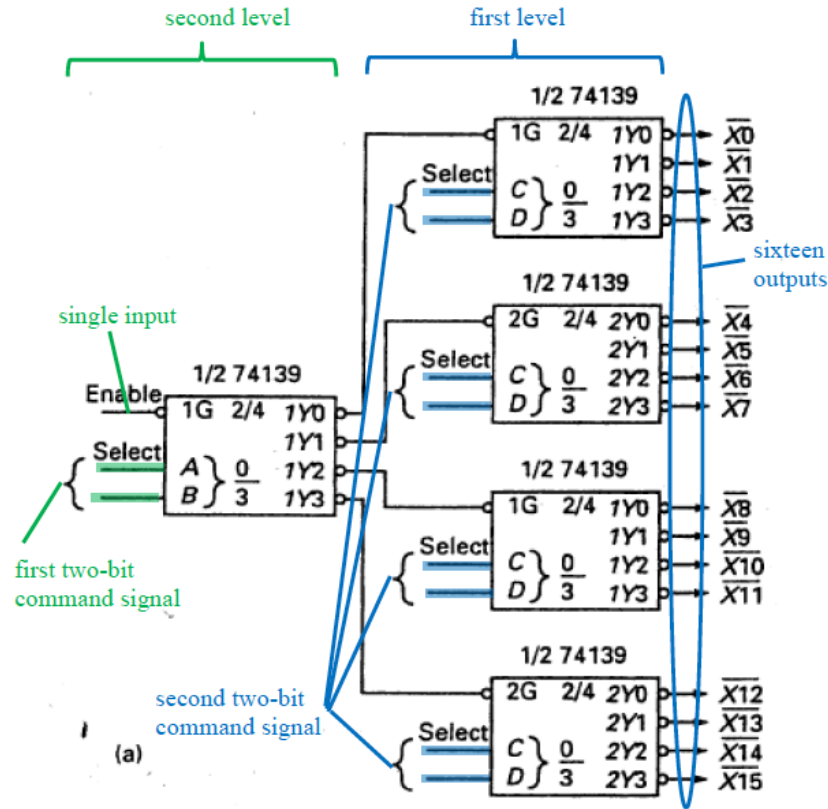
Bruckman, Fig. 2 (annotated); Houh, ¶138.

Bruckman's description of these control lines is limited. Because each of the demultiplexing devices (switching core 40 and concentrators 50) in Fig.2 has one input and four outputs, a POSITA would've found it obvious to control each with a two-bit command signal.¹⁷² A two-bit command signal would efficiently be able to specify one of four operating states, each corresponding to the four possible outputs.¹⁷³ Thus, in Bruckman's example, a two-bit command signal would be used to control the switching core (second-level demultiplexer) to select a trace 52. A different two-bit signal would be used to control the concentrators (first-level demultiplexers) to select a link 30, consistent with known techniques for

¹⁷² *Id.*, ¶139.

¹⁷³ *Id.*

controlling two-level demultiplexers. Such techniques, illustrated below, would've been in the background knowledge of a POSITA.¹⁷⁴



Holdsworth, Fig. 5.17(a) (annotated); Houh, ¶139.

As explained above at [1.6], Bruckman's hashing function produces a single hash value that is used to select both a line from the first level and a line from the second level. In Bruckman's example where there are 16 possible links, the lagSize parameter is 16. The operation "hash % 16" would produce a value ("modulo")

¹⁷⁴ See VI.E; Houh, ¶139.

within the range of 0 and 15.¹⁷⁵ The sixteen different values between 0 and 15 are represented in binary as a four-bit value (“binary representation of the modulo”).

Consistent with a POSITA’s background knowledge of using two two-bit control signals to control a two-level demultiplexer (formed from 1:4 demultiplexers), two of the four hash bits would be used to control the first level and the remaining two of the four hash bits would be used to control the second level.¹⁷⁶ Thus, a POSITA would’ve found it obvious for Bruckman’s control signals (represented by dotted lines in Fig. 2) to be two-bit signals formed from different bits of the hash value.¹⁷⁷

Thus, because Bruckman illustrates control signals to both levels of a two-level demultiplexer, and a POSITA would’ve known the technique for controlling such a structure involves subsets of bits for control of different levels, Bruckman renders obvious “selecting the first and second physical links responsively to respective first and second subsets of bits in a binary representation of the modulo” as claimed.¹⁷⁸

¹⁷⁵ Houh, ¶140.

¹⁷⁶ See Holdsworth, Fig. 5.17(a); Houh, ¶141.

¹⁷⁷ Houh, ¶141.

¹⁷⁸ *Id.*, ¶¶137-42.

Q. Claim 12

For element [12.1], as explained above at [1.5], Bruckman's equipment 22 receives a data frame with frame attributes. As Bruckman's Background notes, it was known to use "information carried in each Ethernet frame in order to make [a] decision as to the physical port to which the frame should be sent.... The information used to assign conversations to ports could thus include one or more of the following pieces of information: a) Source MAC address b) Destination MAC address."¹⁷⁹ Moreover, POSITAs would've recognized that the list of options recited in claim 12 are well-known data fields and that are commonly part of a layer 2 data frame.¹⁸⁰

Thus, because Bruckman's equipment receives a data frame having at least a source MAC address and a destination MAC address, Bruckman renders this limitation obvious.¹⁸¹

R. Claim 13

For the same reasons discussed above, elements [13.0]-[13.5] are obvious over the prior art.¹⁸²

¹⁷⁹ Bruckman, [0005]-[0011].

¹⁸⁰ See *e.g.*, Ex. 1010; Houh, ¶143.

¹⁸¹ Houh, ¶¶143-45.

¹⁸² See [1.0]-[1.7]; Houh, ¶¶146-151.

Regarding element [13.6], first, as explained above at [1.1] and [1.3], Bruckman renders obvious “coupling the network node to the one or more interface modules and coupling each of the one or more interface modules to the communication network.” Second, Bruckman explains that service level agreements between network service providers (including a “communication network”) and customers (including a “customer node”) specify bandwidth in various terms. “In general, the bandwidth guaranteed by a service provider, referred to as the peak information rate (PIR), may include either CIR, or EIR, or both CIR and EIR (in which case $PIR = CIR + EIR$).”¹⁸³

Thus, because Bruckman describes that providing guaranteed bandwidth in terms of either CIR, or EIR, or both CIR and EIR (in which case $PIR = CIR + EIR$), Bruckman renders this limitation obvious.¹⁸⁴

For element [13.7], Bruckman describes allocating bandwidth in a manner so as to exceed the guaranteed bandwidth (“responsively to the bandwidth requirements”).¹⁸⁵ Bruckman describes “a method for establishing a connection with a guaranteed bandwidth for transmitting data between first and second

¹⁸³ Bruckman, [0013].

¹⁸⁴ Houh, ¶¶152-154.

¹⁸⁵ See Bruckman, [0030].

endpoints.”¹⁸⁶ Accordingly, establishing the communication sessions in Bruckman are for the purpose of providing the guaranteed bandwidth and is thus “responsively to the bandwidth requirements.”

Thus, because Bruckman allocates bandwidth to exceed the guaranteed bandwidth, Bruckman renders this limitation obvious.¹⁸⁷

S. Claim 14

Regarding element [14.0], as explained above at [1.0], [1.1], and [1.3], Bruckman describes equipment 22 that connects a plurality of links 30 (which may correspond to the claimed “ports”) to a communication network. Alternatively, Bruckman’s links are associated with ports.¹⁸⁸ The ports associated with links 30 correspond to “user ports” because they connect to user nodes, as explained in [2.2]. Thus, Bruckman renders this limitation obvious.¹⁸⁹

For element [14.1], consistent with the discussion at [1.1] and [14.0], Bruckman renders obvious “coupling the user ports to one or more user interface modules” as claimed.¹⁹⁰

¹⁸⁶ *Id.*, [0027]; *see also id.*, [0016], [0030].

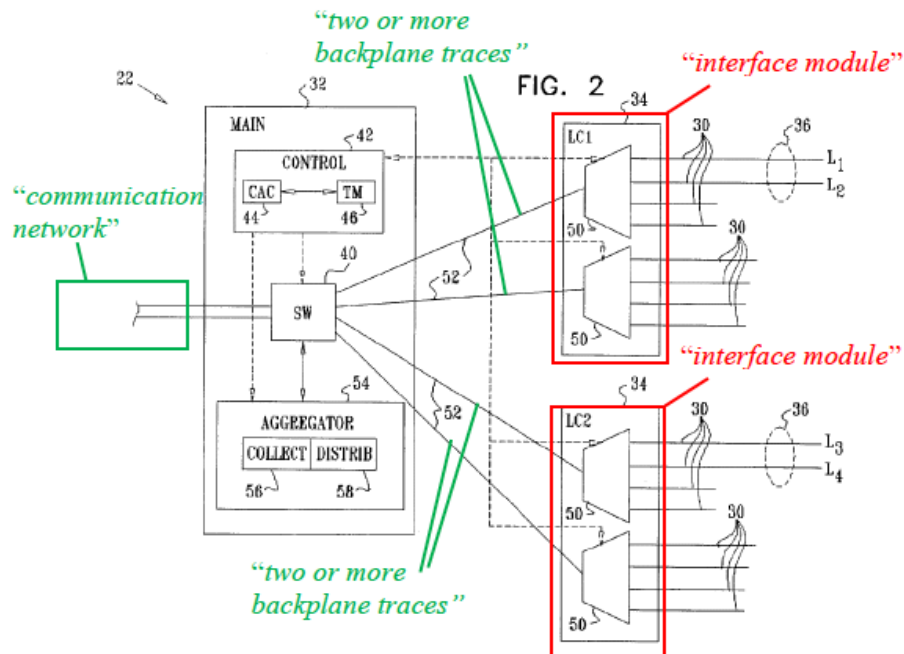
¹⁸⁷ Houh, ¶¶155-157.

¹⁸⁸ Bruckman, [0056] (“multiple ports that serve respective links 30”).

¹⁸⁹ Houh, ¶158.

¹⁹⁰ *Id.*, ¶159.

For element [14.2], first, consistent with the discussion at [1.3], Bruckman renders obvious “coupling each user interface module to the communication network via a backplane” as claimed.¹⁹¹ Second, as shown in Fig. 2 below, each of Bruckman’s two line cards 34 (“interface modules”) includes at least two traces 52 that connect that line card to the communication network.



Bruckman, Fig. 2 (annotated); Houh, ¶161.

Thus, Bruckman renders obvious “coupling each user interface module to the communication network via a backplane using two or more backplane traces arranged in parallel” as claimed.¹⁹²

¹⁹¹ See also [3.2] (explaining that Bruckman’s traces 52 are backplane traces).

¹⁹² Houh, ¶¶160-162.

Regarding element [14.3], as discussed at [1.4], Bruckman renders obvious this limitation.¹⁹³

For elements [14.4]-[14.5], as discussed at [1.6], Bruckman renders obvious this limitation.¹⁹⁴

For element [14.6], as discussed at [1.7], Bruckman renders obvious this limitation.¹⁹⁵

T. Claim 15

For the same reasons discussed above, elements [15.0]-[15.4] are obvious over the prior art.¹⁹⁶

For element [15.5], first, consistent with the discussion above at [1.7], Bruckman renders obvious “sending the data frame over the selected backplane trace” as claimed.¹⁹⁷ Second, Bruckman explains that the traces 52 are part of a link aggregation group: “As a result of spreading group 36 over two (or more) line cards, the link aggregation function applies not only to links 30 in group 36 but

¹⁹³ *Id.*, ¶163.

¹⁹⁴ *Id.*, ¶¶164-165; *see also* [3.1] (explaining Bruckman’s traces 52 are backplane traces).

¹⁹⁵ Houh, ¶166; *see also* [3.1] (explaining Bruckman’s traces 52 are backplane traces).

¹⁹⁶ *See* [14.0]-[14.5]; Houh, ¶¶167-171.

¹⁹⁷ *See also* [3.1] (explaining that Bruckman’s traces 52 are backplane traces).

also to traces 52 that connect to multiplexers 50 that serve these links.”¹⁹⁸ As also explained above at [4.6], Bruckman’s link aggregation groups may be Ethernet link aggregation groups.

Thus, because Bruckman’s equipment 22 sends data over selected traces 52 and links 30, and the traces 52 are part of an Ethernet link aggregation group, Bruckman renders this limitation obvious.¹⁹⁹

U. Claim 16

The prior art discloses element [16.1].²⁰⁰

V. Claim 17

For element [17.0], as explained above at [1.1] and [1.3], Bruckman describes Equipment 22 (“apparatus”) that connects customer nodes (a “network node”) with a communication network. Thus, Bruckman renders this limitation obvious.²⁰¹

Regarding element [17.1], first, Bruckman’s Equipment 22 includes a plurality of line cards 34 (“one or more interface modules”).²⁰²

¹⁹⁸ Bruckman, [0057].

¹⁹⁹ Houh, ¶¶172-74.

²⁰⁰ See claims 8, 9. Houh, ¶175.

²⁰¹ Houh, ¶176.

²⁰² See Bruckman, [0049].

Second, Bruckman's line cards process data frames between a customer node ("network node") and the communication network. "Each line card 34 comprises one or more concentrators 50, which comprise multiple ports that serve respective links 30. The concentrators multiplex data traffic between links 30 and traces 52, which connect the concentrators to switching core 40."²⁰³

Third, consistent with the discussion at [1.5], Bruckman's data frames include frame attributes.

Thus, because Bruckman's equipment includes a plurality of line cards that multiplex traffic from a customer node to a communication network, and the traffic includes data frames with frame attributes, Bruckman renders obvious "one or more interface modules, which are arranged to process data frames having frame attributes sent between the network node and the communication network" as claimed.²⁰⁴

For element [17.2], consistent with the discussion above at [1.2] and [1.4], Bruckman's Equipment processes traffic in both upstream and downstream directions. Accordingly, the line cards ("interface modules") are "operative to

²⁰³ *Id.*, [0056].

²⁰⁴ Houh, ¶¶177-180.

communicate in both an upstream direction and a downstream direction.” Thus, Bruckman renders this limitation obvious.²⁰⁵

For element [17.3], consistent with the discussion above at [1.1], Bruckman renders this limitation obvious.²⁰⁶

For element [17.4], consistent with the discussion above at [1.3], Bruckman renders this limitation obvious.²⁰⁷

Regarding element [17.5], first, Bruckman’s Equipment 22 includes a controller 42 (“control module”): “The operation of switch 40 is managed by a controller 42, typically an embedded microprocessor with suitable software for carrying out the functions described herein.”²⁰⁸ Bruckman further describes additional software components running on controller 42, including “Connection Admission Control entity (CAC) 44,” “traffic manager 46,” and “Aggregator 54.”²⁰⁹

Second, consistent with the discussion above at [1.6], Bruckman’s controller 42 (including aggregator 54) is “arranged to select for each data frame sent

²⁰⁵ *Id.*, ¶181.

²⁰⁶ *Id.*, ¶182.

²⁰⁷ *Id.*, ¶183.

²⁰⁸ Bruckman, [0049].

²⁰⁹ *Id.*, [0050], [0056]-[0057].

between the communication network and the network node, in a single computation based on at least one of the frame attributes, a first physical link out of the first group and a second physical link out of the second group over which to send the data frame” as claimed. Thus, Bruckman (alone or in combination with Basso) renders this limitation obvious.²¹⁰

Regarding element [17.6], the prior art discloses the element.²¹¹

W. Claim 18

Consistent with the discussion above at [3.1], Bruckman renders element [18.1] obvious.²¹²

X. Claim 19

For elements [19.0]-[19.4], the prior art discloses the limitations.²¹³

For element [19.5], the prior art discloses the limitations.²¹⁴

Y. Claim 20

For elements [20.0]-[20.4], the prior art discloses the limitations.²¹⁵

²¹⁰ Houh, ¶¶184-185.

²¹¹ See [1.2], [1.4]; Houh, ¶186.

²¹² Houh, ¶187.

²¹³ See [17.0]-[17.5]; Houh, ¶¶188-192.

²¹⁴ See [4.6]; Houh, ¶193.

²¹⁵ See [17.0]-[17.5]; Houh, ¶¶194-198.

For element [20.5], the prior art discloses the limitations.²¹⁶

Z. Claims 21-27

Claims 21-27 depend (directly or indirectly) from apparatus claim 17 and recite in apparatus form substantially the same subject matter limitations as claims 6-12 (respectively), which depend from method claim 1. Accordingly, claims 21-27 are obvious for the same reasons discussed previously for claims 6-12, respectively.²¹⁷

AA. Claim 28

For elements [28.0]-[28.4], the prior art discloses the claim limitations.²¹⁸

Regarding element [28.5], the prior art discloses the claim limitation.²¹⁹

For element [28.6], consistent with the discussion above at [13.7], Bruckman renders this limitation obvious. A POSITA would've understood that if the links can provide the allocated bandwidth, they are thus "dimensioned" to provide such bandwidth. The links would otherwise not be able to provide the required bandwidth.²²⁰

²¹⁶ See [5.6], [5.7]; Houh, ¶199.

²¹⁷ Houh, ¶¶200-210.

²¹⁸ See [17.0]-[17.5]; Houh, ¶¶211-215.

²¹⁹ See [13.6]; Houh, ¶216.

²²⁰ Houh, ¶217.

BB. Claim 29

For elements [29.0]-[29.2], the prior art discloses the limitation.²²¹

For element [29.3], consistent with the discussion above at [3.1] and [18.1], Bruckman renders this limitation obvious.²²²

Regarding element [29.4], consistent with the discussion above at [1.2] and [1.4], Bruckman renders this limitation obvious.²²³

For element [29.5], the prior art discloses the limitation.²²⁴

CC. Claim 30

Regarding element [30.0], the prior art discloses this limitation.²²⁵

For elements [30.1]-[30.2], consistent with the discussion above at [17.1], Bruckman renders this limitation obvious.²²⁶

For element [30.3], the prior art discloses this limitation.²²⁷

For element [30.4], the prior art discloses this limitation.²²⁸

²²¹ See [17.0]-[17.2]; Houh, ¶¶218-220.

²²² Houh, ¶221.

²²³ *Id.*, ¶222.

²²⁴ See [19.4]; Houh, ¶223.

²²⁵ See [14.0], [17.0]; Houh, ¶224.

²²⁶ Houh, ¶225.

²²⁷ See [29.3]; Houh, ¶226.

²²⁸ See [19.4]; Houh, ¶227.

Regarding element [30.5], the prior art discloses this limitation.²²⁹

DD. Claim 31

Regarding element [31.1], the prior art discloses this limitation.²³⁰

**VII. GROUNDS 3 AND 4: BRUCKMAN (ALONE OR WITH BASSO) IN VIEW OF
HOLDSWORTH RENDER OBVIOUS CLAIMS 11 AND 26.**

Grounds 1 and 2 show how Bruckman renders obvious limitations related to using different subsets of bits to select different levels in a multi-level demultiplexer. Grounds 1 and 2 further rely on the Holdsworth textbook as evidence of the background knowledge of a POSITA. To the extent Patent Owner argues that such digital logic design wouldn't have been within the background knowledge of a POSITA, Grounds 3 and 4 show that combining the disclosures of Holdsworth with the other prior art would've been obvious to a POSITA.²³¹

A. Reasons to Combine Holdsworth with Bruckman

A POSITA would've found it obvious to use basic digital logic design to implement the control signals for Bruckman's two-level demultiplexer. For example, a POSITA would've found it obvious to use a first subset of bits for a

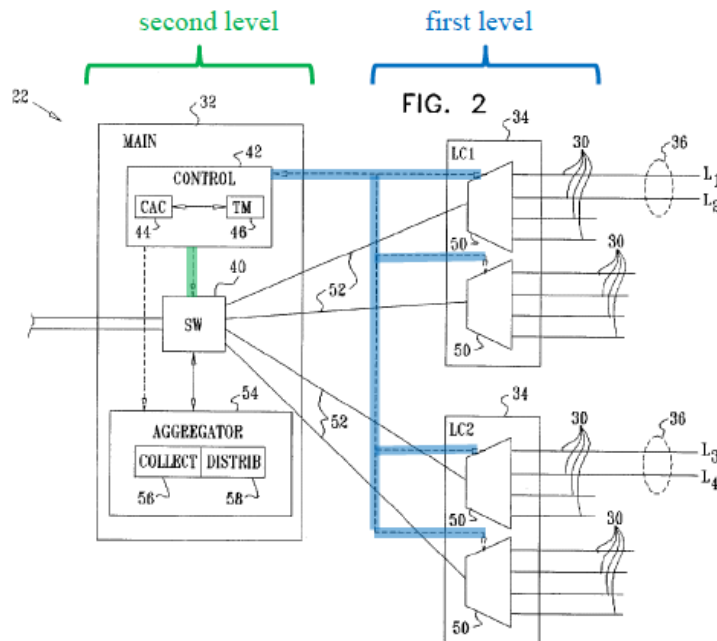
²²⁹ See [15.5]; Houh, ¶228.

²³⁰ See [8.1]; Houh, ¶229.

²³¹ Houh, ¶230.

first-level demultiplexer, and a second subset of bits for the second-level demultiplexers, as evidenced by Holdsworth.²³²

As explained above, Bruckman's Equipment 22 uses a two-level demultiplexer structure for distributing incoming packets from the communication network among the links 30. Bruckman's depiction of two-level demultiplexer includes dotted lines from the controller to the first-level and second-level demultiplexers, as shown in Fig. 2 below.



Bruckman, Fig. 2 (annotated); Houh, ¶232.

A POSITA would've recognized that these dotted lines represent command signals for controlling each of the demultiplexers (switching core 40 and

²³² *Id.*, ¶231.

concentrators 50).²³³ Bruckman illustrates these control lines but provides few implementation details about the command signals used on the control lines, indicating that such implementation would've been within the knowledge and skill of a POSITA.²³⁴ Accordingly, a POSITA looking at the teachings of Bruckman would've looked to known implementation techniques for the command signals of Bruckman's two-level demultiplexer.²³⁵ Bruckman explains that "at least some of the functions of the aggregator may be carried out by hard-wired logic or by a programmable logic component, such as a gate array."²³⁶ Accordingly, a POSITA would've looked to known logic design techniques.²³⁷

Holdsworth represents basic logic design that is applicable to Bruckman's two-level demultiplexer structure. As stated in the preface, Holdsworth is intended to "act as a reference text for graduates working in this field."²³⁸ A POSITA would've had a reasonable expectation of success when using Holdsworth's technique because Holdsworth provides an example that is similar to that of Bruckman's example—a plurality of 1:4 demultiplexers that form a 1:16

²³³ *Id.*, ¶233.

²³⁴ *Id.*, ¶233.

²³⁵ *Id.*, ¶233.

²³⁶ Bruckman, [0057].

²³⁷ Houh, ¶233.

²³⁸ Holdsworth, Preface.

demultiplexer. A POSITA would've therefore expected that using a two-bit command signal to control a first-level demultiplexer, and a second two-bit command signal to control second-level demultiplexers would've worked as well in Bruckman's two-level demultiplexer.²³⁹

Given that Bruckman's hash function would produce a hash value in the range of zero to fifteen (and thus be readily represented as a four-bit value), a POSITA would've found it obvious to use two of the four bits to control the second-level demultiplexer (i.e., the switching core) and the other two of the four bits to control the first-level demultiplexers (i.e., the concentrators).²⁴⁰

Regarding Ground 4, application of Holdsworth technique is also consistent with Basso's technique of using the single hash computation to select both a second-level line (one of the traces 52) and a first-level line (one of the links 30). This would've yielded the predictable result of effectively and efficiently controlling both levels of the two-level demultiplexer using the four-bit hash computation described by Bruckman.

Thus, the combination of Holdsworth with Bruckman represents application of a known technique (Holdsworth's different bit subsets for different levels of a two-level demultiplexer) to a known method (Bruckman's two-level

²³⁹ *Id.*, ¶234.

²⁴⁰ *Id.*, ¶235.

demultiplexer) to yield predictable results (controlling both levels with a single hash computation).²⁴¹

B. Claims 11, 26

The claim analysis for this ground is substantially the same as the analysis provided in Grounds 1 and 2.²⁴²

VIII. CONCLUSION

For at least the above reasons, Juniper requests cancelation of claims 1-31 of the '740 Patent.

Respectfully submitted,

/s/ Kyle K. Tsui

Kyle K. Tsui (Reg. No. 62,602)

Phone: 202.362.3527

Email: kyle.tsui@fischllp.com

Andrew L. Ramos (Reg. No. 76,865)

Phone: 202.362.3522

Email: Andrew.Ramos@fischllp.com

FISCH SIGLER LLP

5301 Wisconsin Avenue NW

Suite 400

Washington, DC 20015

²⁴¹ *Id.*, ¶¶231-236.

²⁴² *See supra* VI.P, VI.Z; Houh, ¶237.

CERTIFICATE OF COMPLIANCE

Per 37 C.F.R. §42.24(a) and (d), the undersigned hereby certifies that the
Petition complies with the type-volume limitation of 37 C.F.R. §42.24(a)(i)
because, exclusive of exempted portions, it contains 11,104 words as counted by
the word-processing program used to prepare it.

Dated: October 10, 2023

Respectfully submitted,

/s/ Kyle K. Tsui

Kyle K. Tsui (Reg. No. 62,602)

kyle.tsui@fischllp.com

FISCH SIGLER LLP

5301 Wisconsin Avenue NW

Suite 400

Washington, DC 20015

Phone: 202.362.3527

CERTIFICATE OF SERVICE

The undersigned certifies service pursuant to 37 C.F.R. §§42.6(e) and 42.105(b) on the Patent Owner by Federal Express of a copy of this Petition for Inter Partes Review and supporting materials to the correspondence address of record for U.S. Patent No. 7,545,740:

May Patents Ltd. c/o Dorit Shem-Tov
P.O.B 7230
Ramat-Gan, 5217102
Israel

An electronic courtesy copy is also being served to litigation counsel at:

Jacob R. Kirkham
Kobre & Kim LLP
600 North King Street, Suite 501
Wilmington, DE 19801
jacob.kirkham@kobrekim.com

Dated: October 10, 2023

Respectfully submitted,

/s/ Kyle K. Tsui
Kyle K. Tsui (Reg. No. 62,602)
kyle.tsui@fischllp.com
FISCH SIGLER LLP
5301 Wisconsin Avenue NW
Suite 400
Washington, DC 20015
Phone: 202.362.3527