

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

OXYLABS LT, UAB,
Petitioner,

v.

BRIGHT DATA LTD.,
Patent Owner.

Case IPR2024-00126
Patent 11,616,826

PETITION FOR *INTER PARTES* REVIEW UNDER 35 U.S.C. § 312

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	MANDATORY NOTICES	2
A.	Real Party in Interest (37 C.F.R. § 42.8(b)(1))	2
B.	Related Matters (37 C.F.R. § 42.8(b)(2)).....	3
1.	Litigation.....	3
2.	Administrative Proceedings	4
C.	Lead and Back-Up Counsel (37 C.F.R. § 42.8(b)(3)).....	6
D.	Service Information (37 C.F.R. § 42.8(b)(4))	6
III.	GROUND FOR STANDING.....	7
IV.	THE GROUNDS IN THIS PETITION ARE NOT CUMULATIVE.....	7
V.	STATEMENT OF PRECISE RELIEF REQUESTED FOR EACH CLAIM CHALLENGED	8
A.	Claims for Which Review Is Requested (37 C.F.R. § 42.104(b)(1)): 1-10, 12, 15-19, and 21-25	8
B.	Statutory Grounds of Challenge (37 C.F.R. § 42.104(b)(2))	8
VI.	LEVEL OF ORDINARY SKILL IN THE ART	10
VII.	FIELD OF TECHNOLOGY	10
VIII.	THE '826 PATENT	15
A.	Overview	15
B.	Prosecution History	17
C.	Claim Construction (37 C.F.R. § 42.104(b)(3)).....	17
1.	“Client Device” and “First Server”	17
2.	“Server operating system”	19
3.	“selecting, an IP address” and “sending of the selected IP addresses to the first client device”	20

IX.	REASONS FOR THE RELIEF REQUESTED UNDER 37 C.F.R. §§ 42.22(a)(2) and 42.104(b)(4)	21
A.	Ground 1 – Kocherlakota Renders Obvious Claims 1-5, 10, 12, 15-18, 19, and 21-25	21
1.	Kocherlakota	21
2.	Element-by-Element Analysis	27
B.	Ground 2 – Kocherlakota in View of RFC 1122 Renders Obvious Claims 6-9, ...	47
1.	RFC 1122 Overview	47
2.	Motivation to Combine	48
3.	Modification of Kocherlakota in view of RFC 1122	49
4.	Element-by-Element Analysis	50
C.	Ground 3– Cohen Renders Obvious Claims 1-6, 12, 15-17, 18, 19, and 21-25	53
1.	Cohen	53
2.	Element-by-Element Analysis	56
D.	Ground 4 – Cohen in View of RFC 1122 Renders Obvious Claims 6-9, ...	74
1.	Motivation to combine	74
2.	Modification of Cohen in view of RFC 1122	76
3.	Element-by-Element Analysis	77
X.	CONCLUSION	79

PETITIONER'S EXHIBIT LIST

Ex. #	Shorthand	Description
1001	'826 Patent	U.S. Patent No. 11,616,826
1002	Olivier	Declaration of Dr. James Olivier
1003	Kocherlakota	U.S. Patent No. 6,785,705 (issued Aug. 31, 2004)
1004	Cohen	U.S. Patent No. 6,389,462 (issued May 14, 2002)
1005	RFC 2616	Internet Engineering Task Force, <i>Hypertext Transfer Protocol – HTTP/1.1</i> , June 1999
1006	RFC 1122	Internet Engineering Task Force, <i>Requirements for Internet Hosts – Communication Layers</i> , October 1989
1007	'826 FH	Prosecution History of U.S. Patent No. 11,616,826
1008	'341 FH	Prosecution History of U.S. Patent No. 11,044,341
1009		Reserved
1010	<i>Teso</i> CC Order	Claim Construction Opinion and Order, Document 191, <i>Luminati Networks Ltd. v. Teso LT et al.</i> , Case No. 2:19-cv-00395 (E.D. Tex.)
1011	<i>Code200</i> CC Order	Claim Construction Memorandum Opinion and Order, Document 97, <i>Luminati Networks Ltd. v. Code200 et al.</i> , Case No. 2:19-cv-00396 (E.D. Tex.)
1012	Motion for Hearing	Defendants' Motion for Hearing Regarding O2 Micro Issue, Document 444, <i>Bright Data Ltd. v. Teso LT et al.</i> , Case No. 2:19-cv-00395 (E.D. Tex.)
1013	Supplemental CC Order	Supplemental Claim Construction Order, Document 453, <i>Bright Data Ltd. v. Teso LT et al.</i> , Case No. 2:19-cv-00395 (E.D. Tex.)
1014		Reserved
1015	RFC 791	Internet Engineering Task Force, Request for Comments 1122 <i>Internet Protocol</i> , September 1981
1016	RFC 793	Internet Engineering Task Force, Request for Comments 1122 <i>Transmission Control Protocol</i> , September 1981
1017	RFC 1123	Internet Engineering Task Force, Request for Comments 1123 <i>Requirements for Internet Hosts - Application and Support</i> , October 1989
1018	Coley	U.S. Patent 5,826,014 (issued Oct. 20, 1998)
1019	Haverstock	U.S. Patent No. 6,449,640 (issued Sep. 10, 2002)
1020	RFC Index	August 3, 2001 archived Internet Society webpage for RFC Index from Internet Archive (Wayback Machine),

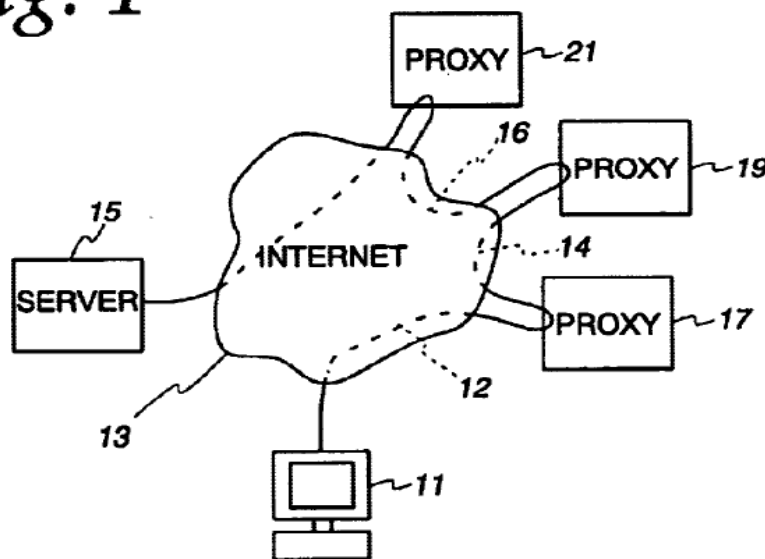
Ex. #	Shorthand	Description
		available at: http://web.archive.org/web/20010803120225/http://www.rfc-editor.org:80/rfc-index2.html
1021	Keepalive HowTo	June 11, 2008 archived Internet Society webpage for TCP KEEPALIVE HOWTO, available at https://web.archive.org/web/20080611153148/https://tldp.org/HOWTO/TCP-Keepalive-HOWTO/index.html
1022	Masters	U.S. Patent No. 8,108,554 to Masters (filed Oct. 27, 2008)
1023	Definitive Guide	Gourley, D., et al. (2002). HTTP: The Definitive Guide. Germany: O'Reilly Media, Incorporated. Available at https://archive.org/details/httpdefinitivegu00gour/page/n9/mode/2up

I. INTRODUCTION

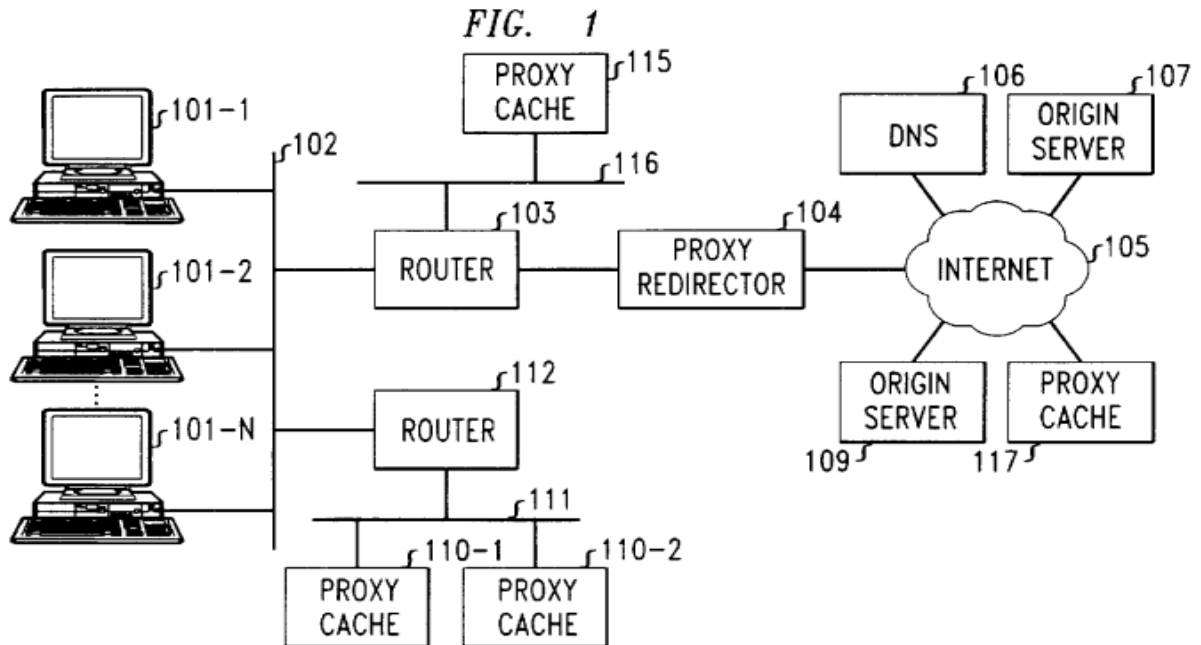
The Challenged Claims recite a “first server,” while acting as a proxy (forwarding web requests onward and returning content to the requester), selecting an IP address in response to receiving a URL, and using that IP address in sending the URL onward. These concepts were well known by 2009.

The '826 Patent acknowledges that proxies were “prior art.” EX1001, 3:66-67. And, each of Kocherlakota and Cohen disclose a proxy analyzing a received URL and, in response, selecting the IP address of downstream device for forwarding. Kocherlakota's FIG. 1, below, discloses proxy 17 receiving URLs from client computer 11 and selecting the IP address for forwarding to proxy 19 or proxy 21 for further forwarding to web server 15 based on the URL. EX1003, 4:15-52, 5:18-24.

Fig. 1



Likewise, Cohen's FIG. 1, below, discloses "proxy redirector 104" receiving web requests from "client 101-1" and "select[ing] one of the available proxy caches" to forward to for further forwarding to an origin (web) server "based ... IP header information such as the origin server IP address." EX1004, 7:36-48.



II. MANDATORY NOTICES

A. Real Party in Interest (37 C.F.R. §42.8(b)(1))

The real parties-in-interest are the Petitioners Oxylabs, UAB; coretech lt, UAB; and Code200, UAB. As of November 30, 2022, oxysales, UAB and metacluster lt, UAB have been merged into Teso LT, UAB, with Teso LT, UAB succeeding to all of the rights and obligations of oxysales, UAB, and metacluster lt, UAB. As of November 30, 2022, Teso LT, UAB began operating as Oxylabs, UAB.

B. Related Matters (§42.8(b)(2))

1. Litigation

The '826 Patent is a continuation of U.S. Patent No. 11,044,341 (the '341 Patent), which is a continuation of U.S. Patent No. 10,469,628 ('628 Patent), which is a continuation of U.S. Patent No. 10,225,374 ('374 Patent), which is a continuation of U.S. Patent No. 10,069,936 ('936 Patent) ("Parent Patents").

Through these Parent Patents, the '826 Patent shares a common specification and priority claim with U.S. Patent Nos. 10,484,511 (the '511 Patent), 11,190,622 (the '622 Patent); 10,257,319 (the '319 Patent); 10,484,510 (the '510 Patent); and 10,637,968 (the '968 Patent). PO has accused Oxylabs (as Code200, Metaccluster, and Oxysales) of infringing claims of the '511 Patent and the '968 Patent in *Bright Data Ltd. v. Code200, UAB*, Case No. 2:19-cv-00396 (E.D. Tex.) (the "*Code200* Litigation"). PO has also accused Oxylabs (as Teso, Metaccluster, and Oxysales) of infringing claims of the '319 and '510 Patents in *Bright Data Ltd. v. Teso LT, UAB*, Case No. 2:19-cv-00395 (E.D. Tex.) ("*Teso* Litigation"). Together, these actions are the "Related Litigation," and the patents asserted in the Related Litigation are referred to in this Petition as the "Related Patents."

There is no litigation involving the '826 Patent. The table below lists litigation involving patents claiming common priority to the '826 Patent.

Cases
<i>Bright Data Ltd. v. Teso LT, UAB et al.</i> , Case No. 2:19-CV-00395 (E.D. Tex.).
<i>Bright Data Ltd. v. NordVPN</i> , Case No. 2:19-CV-00414 (E.D. Tex.)
<i>Bright Data Ltd. v. NetNut, Ltd.</i> , Case No. 2:21-CV-00225 (E.D. Tex.)
<i>Luminati Networks Ltd. v. BI Science (2009) Ltd.</i> , Case No. 2:19-CV-00397 (E.D. Tex.) (closed)
<i>Bright Data Ltd. v. Code200, UAB et al.</i> , Case No. 2:19-CV-00396 (E.D. Tex.)
<i>Luminati Networks Ltd. v. NetNut, Ltd.</i> , Case No. 2:20-CV-00188 (E.D. Tex.) (closed)

2. Administrative Proceedings

The Related Patents were the subject of petitions for *inter partes* review in IPR2020-01266, IPR2020-01358, IPR2021-00122, and IPR2021-00249. The Board exercised its discretion and declined to institute review based on the *Fintiv* factors – the Board considered, but did not discuss any specific aspect of, the substantive merits of each petition within that framework. *Code200 v. Luminati Networks, Ltd.*, IPR2020-01266, Paper 18 at 11-12 (PTAB Dec. 23, 2020); *Code200 v. Luminati Networks, Ltd.*, IPR2020-01358, Paper 11 at 11 (PTAB Feb. 2, 2021); *Teso LT, UAB v. Luminati Networks Ltd.*, Case No. IPR2021-00122, Paper 12 at 11 (PTAB April 20, 2021); *Teso LT, UAB v. Luminati Networks Ltd.*, Case No. IPR2021-00249, Paper 12 at 12 (PTAB Feb. 6, 2021);

Additionally, the Related Patents are also the subject of additional IPR petitions. The table below lists IPR petitions for patents claiming common priority to the '826 Patent.

IPR Proceeding	Subject Matter	Status
Case No. IPR2020-01506	US 10,469,614	Institution Denied - Procedural
Case No. IPR2021-00122	'511 Patent	Institution Denied - Procedural
Case No. IPR2021-00249	'968 Patent	Institution Denied - Procedural
Case No. IPR2020-01266	'319 Patent	Institution Denied - Procedural
Case No. IPR2020-01358	'510 Patent	Institution Denied - Procedural
Case No. IPR2021-01502	US 9,241,044	Institution Denied - Procedural
Case No. IPR2021-01503	US 9,742,866	Institution Denied - Procedural
Case No. IPR2022-00103	US 11,044,342	All Challenged Claims Held Unpatentable
Case No. IPR2022-00135	'319 Patent	All Challenged Claims Held Unpatentable
Case No. IPR2022-00138	'510 Patent	All Challenged Claims Held Unpatentable
Case No. IPR2022-00353	US 11,044,344	All Challenged Claims Held Unpatentable
Case No. IPR2022-00687	'622 Patent	Instituted
Case No. IPR2022-00861	'319 Patent	Instituted
Case No. IPR2021-01492	'319 Patent	Joined with IPR2022-00861
Case No. IPR2022-00862	'510 Patent	Instituted
Case No. IPR2021-01493	'510 Patent	Joined with IPR2022-00862
Case No. IPR2022-00936	US 11,206,317	Institution Denied
Case No. IPR2022-01109	'319 Patent	Institution Denied - Procedural
Case No. IPR2022-01110	'510 Patent	Institution Denied - Procedural

C. Lead and Back-Up Counsel (§42.8(b)(3))

Lead Counsel	Back-Up Counsel
Eagle H. Robinson, Reg. No. 61,361	Daniel S. Leventhal, Reg. No. 59,576 Chad Wallis, Reg. No. 78,733

D. Service Information (§42.8(b)(4))

Lead Counsel	Back-Up Counsel
Eagle H. Robinson Norton Rose Fulbright US LLP 98 San Jacinto Blvd., Suite 1100 Austin, TX 78701 512.536.3083 (telephone) 512.536.4598 (facsimile) eagle.robinson@nortonrosefulbright.com	Daniel S. Leventhal Norton Rose Fulbright US LLP Fulbright Tower 1301 McKinney, Suite 5100 Houston, TX 77010-3095 713.651.5151 (telephone) 713.651.5246 (facsimile) daniel.leventhal@nortonrosefulbright.com Chad Wallis Norton Rose Fulbright US LLP 98 San Jacinto Blvd., Suite 1100 Austin, TX 78701 512.474.5201 (telephone) 512.536.4598 (facsimile) chad.wallis@nortonrosefulbright.com

Petitioners consent to electronic service.

III. GROUNDS FOR STANDING

Pursuant to 37 C.F.R. § 42.104(a), Petitioners certify that the '826 Patent is available for *inter partes* review¹, and Petitioners are not barred or estopped from requesting an IPR on the grounds identified herein. The '826 Patent has not been subject to a previous final written decision in an estoppel-based AIA proceeding.

IV. THE GROUNDS IN THIS PETITION ARE NOT CUMULATIVE

Kocherlakota, Cohen, and RFC 1122 are among over 500 references submitted in IDSs during prosecution, but none were substantively discussed by the Examiner. EX1007, 48-51, 104-115. The Examiner did not issue any rejections beyond a double-patenting rejection over the '341 Patent, the '374 Patent, and the '628 Patent. *Id.*, 104-115. As set forth below, the Examiner erred by allowing the claims over the Grounds in this Petition, and the lack of analysis during prosecution weighs against discretionary denial. *Advanced Bionics, LLC v. MED-EL Elektromedizinische Geräte Ges.m.b.H et al.*, No. IPR2019-01469, Paper 6 at 10-11 (P.T.A.B. Feb. 13, 2020).

Moreover, the Grounds in this Petition are not cumulative of the examination of the Parent Patents. For example, in the '341 Patent, Applicants overcame a Yu-

¹ The '826 Patent includes pre-AIA priority claims, which Petitioners are not challenging here.

Barth obviousness rejection by arguing non-combinability and that the mapped prior art “first content” sent to the client device is a manipulation of the user request rather than content requested or addressed by the user. EX1008, 124-138. Applicant further argued that the combination did not perform the claimed “selecting” step. *Id.* The Grounds here do not rely on Yu or Barth, and each of Kocherlakota and Cohen disclose the mapped “first content” and performing the “selecting” step. Confirming the Grounds are not cumulative over prior examination, the ’511 Patent—a Related Patent—is undergoing reexamination, and the CRU has determined that each of Kocherlakota and Cohen raise a SNQ. EX1009, 15-20/24.

V. STATEMENT OF PRECISE RELIEF REQUESTED

A. Claims for Which Review Is Requested (§42.104(b)(1)):

Petitioner requests cancellation of claims 1-10, 12, 15-19, and 21-25.

B. Statutory Grounds of Challenge (§42.104(b)(2))²

Ground 1: Claims 1-5, 10, 12, 15-19, and 21-25 are invalid under § 103(a) over U.S. Pat. No. 6,785,705 (“Kocherlakota”), which issued on August 31, 2004, and is prior art under § 102(b).

² All Grounds are supported by a POSITA’s general knowledge. *Koninklijke Philips N.V. v. Google LLC*, 948 F.3d 1330, 1337-38 (Fed. Cir. 2020).

Ground 2: With Ground 1 as predicate, claims 6-9 are invalid under § 103(a) over Kocherlakota in view of RFC 1122³ and knowledge of a POSITA. RFC 1122 was published twenty years before the alleged priority date by the Network Working Group of the Internet Engineering Task Force (IETF) in October 1989 and is prior art under § 102(b). EX1006, 1/116. RFCs posted on the Internet are published in the ordinary course by established standards organizations and are intended to be viewed by the interested Internet engineering audience at large as of their dates of publication as stated on the cover of each. EX1002, ¶86.

Ground 3: Claims 1-6, 10, 12, 15-19, and 21-25 are invalid under § 103(a) over U.S. Pat. No. 6,389,462 (“Cohen”), which issued May 14, 2002, and is prior art under § 102(b).

³ RFC 1122 was obtained from <https://datatracker.ietf.org/doc/html/rfc1122> (EX1002, ¶87), and was publicly indexed no later than 2001. EX1020, 76/109, 1/109 (RFC Index archived Aug. 3, 2001). Further, RFC 1122 is the companion specification to RFC 1123 titled “Requirements for Internet Hosts -- Application and Support” that also published in October 1989. EX1017, 1/98; EX1002, ¶86; EX1020, 76/109. RFC 1123 is cited in RFC 2616, which the ’826 Patent confirms as prior art. EX1005, 97/114; EX1001, 16:21-22; EX1020, 20/109.

Ground 4: With Ground 3 as predicate, claims 6-9 are invalid under § 103(a) over Cohen in view of RFC 1122 and knowledge of a POSITA.

VI. LEVEL OF ORDINARY SKILL IN THE ART

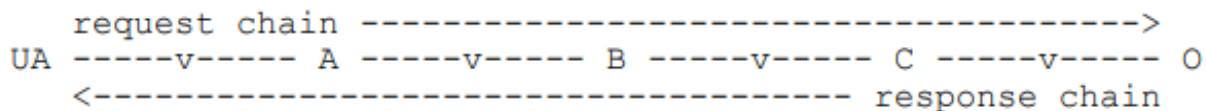
A POSITA as of October 8, 2009⁴—the '826 Patent's earliest claimed priority date—would have had at least a bachelor's degree in computer science or related field (or equivalent experience), and at least two years' experience working with and programming networked computer systems. EX1002, ¶33. The prior art and the '826 Patent also evidence this level of ordinary skill. Here, the background technology described in Section VII and the prior art described in Section IX demonstrate that a POSITA would have been familiar with the underlying principles of Web, Internet, network communication, data transfer, and content sharing across networks, including the HTTP and TCP/IP protocols. *Id.*, ¶¶33-34, 36.

VII. FIELD OF TECHNOLOGY

Web communications and routing were well-known prior to the priority date. RFC 2616, which the '826 Patent recognizes “define[s]” “the HTTP protocol,” was

⁴ All statements in this Petition about the knowledge and skills of, and what would have been obvious to, a POSITA are offered from this perspective as of this date.

published a decade earlier in 1999.⁵ EX1001, 16:21-22; EX1005, 1/114; EX1002, ¶40. In addition to describing “client” and “server” operations (*see* Section VIII.C.1), RFC 2616 discloses multi-hop routing using “proxy” devices running an “intermediary program which acts as both a server and a client for the purpose of making requests on behalf of other clients.” EX1005, 8/114. That is, while communication may be “accomplished via a single connection (v) between the user agent (UA) and the origin server (O),” “more complicated situation[s]” were also known “when one or more intermediaries are present in the request/response chain,” such as an illustrated chain with “three intermediaries (A, B, and C).”



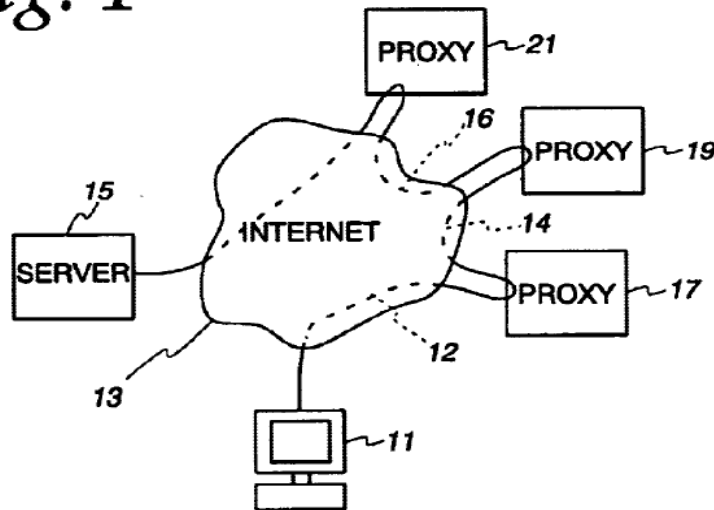
Id., 10/114; EX1002, ¶42.

It was also well-known for an intermediary to determine routing of a received request based on that request. EX1002, ¶43. For example, Kocherlakota discloses proxy 17 receiving URLs from client computer 11 and selecting the IP address for

⁵ RFC 2616 was published by Network Working Group of the Internet Engineering Task Force (IETF) in June 1981. EX1005, 1/114. RFC 2616 was obtained from https://datatracker.ietf.org/doc/html/rfc2616__(EX1002, ¶40) and was publicly indexed by 2001. EX1020, 20/109, 1/109 (RFC Index archived Aug. 3, 2001).

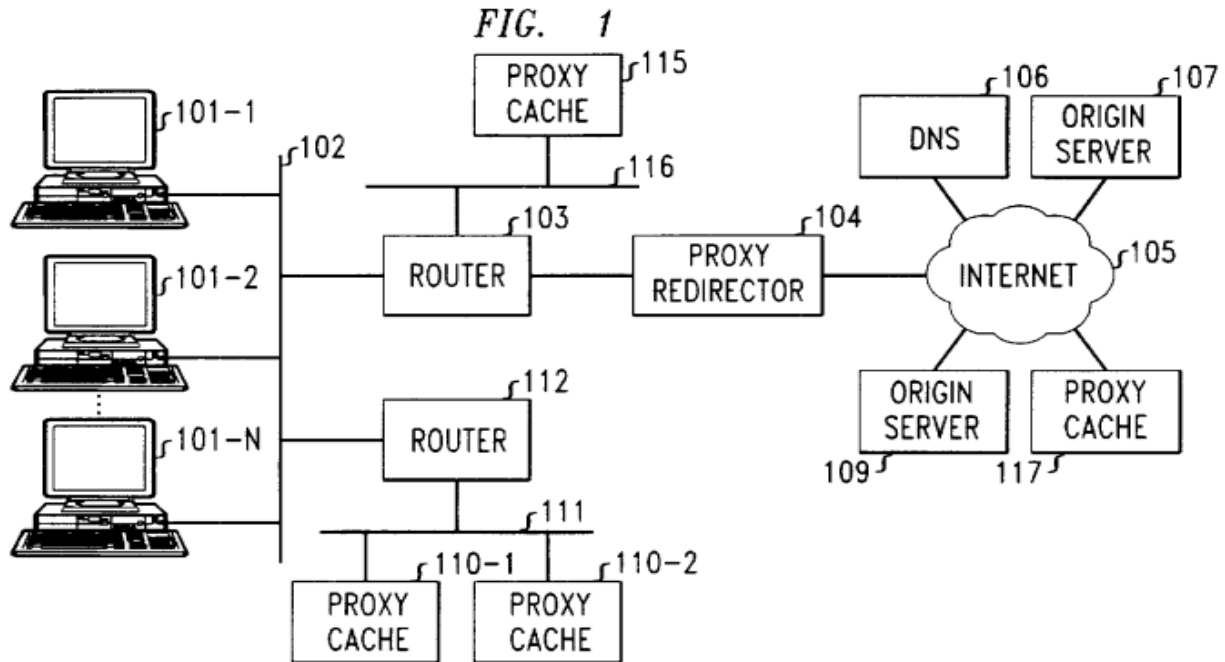
forwarding to proxy 19 or proxy 21 for further forwarding to web server 15 based on the URL. EX1003, 4:15-52, 5:18-24; EX1002, ¶43.

Fig. 1



EX1003, FIG. 1.

Likewise, Cohen discloses “proxy redirector 104” intercepting web requests from “client 101-1” and “select[ing] one of the available proxy caches” for further forwarding to an origin (web) server “based on IP header information such as the origin server IP address.” EX1004, 7:36-48; EX1002, ¶44.



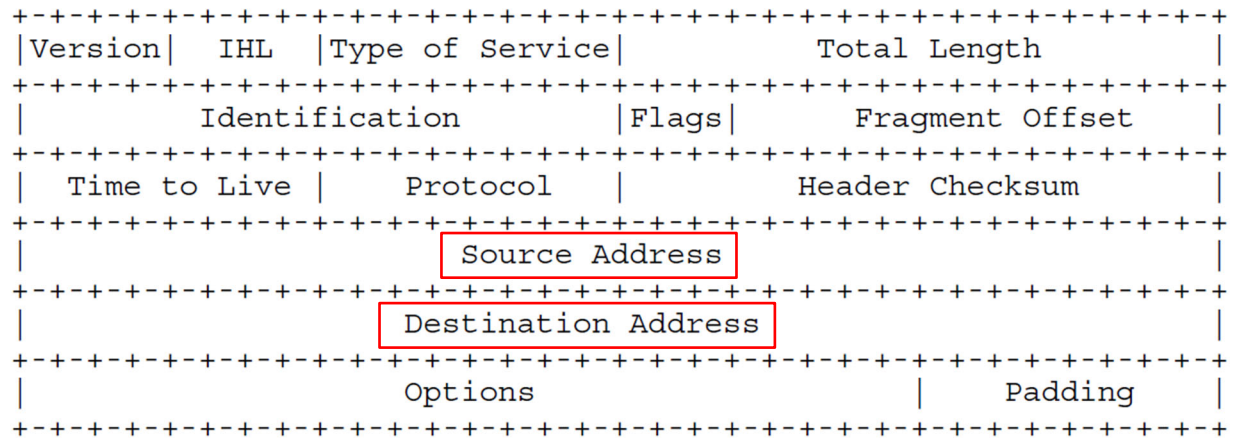
EX1004, FIG. 1.

HTTP's relationship to other well-known Internet standards was also known, including RFC 791⁶ and RFC 793.⁷ EX1002, ¶¶46, 49; *see also* EX1005, 11/114

⁶ RFC 791 was published by Network Working Group of the Internet Engineering Task Force (IETF) in September 1981. EX1015, 1. RFC 791 was obtained from <https://datatracker.ietf.org/doc/rfc791/> (EX1002, ¶46) and was publicly indexed no later than 2001. EX1020, 87/109, 1/109 (RFC Index archived Aug. 3, 2001).

⁷ RFC 793 was published by Network Working Group of the Internet Engineering Task Force (IETF) in September 1981. EX1016, 1. RFC 793 was obtained from <https://datatracker.ietf.org/doc/rfc793/> (EX1002, ¶49) and was publicly indexed no later than 2001. EX1020, 87/109, 1/109 (RFC Index archived Aug. 3, 2001).

(disclosing that “HTTP communication usually takes place over TCP/IP connections”); EX1001, 17:20-26 (describing “TCPIP” as “known” and referencing a “Web server” that “is a TCPIP server”). As annotated in the figure below, IP Packets include a header that includes a “source address” and “destination address”:



Example Internet Datagram Header

Figure 4.

EX1015, 17/51; EX1016, 57/91 (in TCP, “[a]ny lower level protocol will have to provide the source address, destination address”); EX1002, ¶47.

Advanced features of TCP were also known, notably the use of a “‘keep-alive’ mechanism” “[t]o confirm that an idle connection is still active.” EX1006, 102/116; EX1002, ¶52. It was well-known to a POSITA how to implement such as keep alive mechanism. EX1002, ¶53. One example from the well known “The Linux Documentation Project” HOWTO guides illustrates one way a POSITA would have implemented a TCP keep alive in a Linux operating system. EX1002, ¶53. Such

implementations would have included sending probe packets, setting up a countdown timer, and sending problem packets . EX1002, ¶53. A probe packet will be sent to a device and, if a reply is received before the timer reaches zero, then the connection is assumed to be alive. EX1002, ¶53; EX1021, §2.1. If a reply is not received before the timer reaches zero, then a count is kept of the unacknowledged probes; if this count exceeds a value the connection is considered dead. EX1002, ¶53; EX1021, §3.

VIII. THE '826 PATENT

A. Overview

The claims of the '826 Patent bear little resemblance to the specification. The specification discloses use of caching proxies to “improv[e] data communication speed and bandwidth efficiency on the Internet.” EX1001, 1:24–25. The system comprises an acceleration server and a group of identical devices with multiple modules to act in the role of a client, peer, or agent. *Id.*, 4:43-4:61, 5:8-48, 5:53-55, 9:18-50, 12:33-56, FIGs. 3-6. FIG. 3 labels these identical devices with client, agent, and peer labels based on the role the device performs when the web browser on client 102 makes a web request.

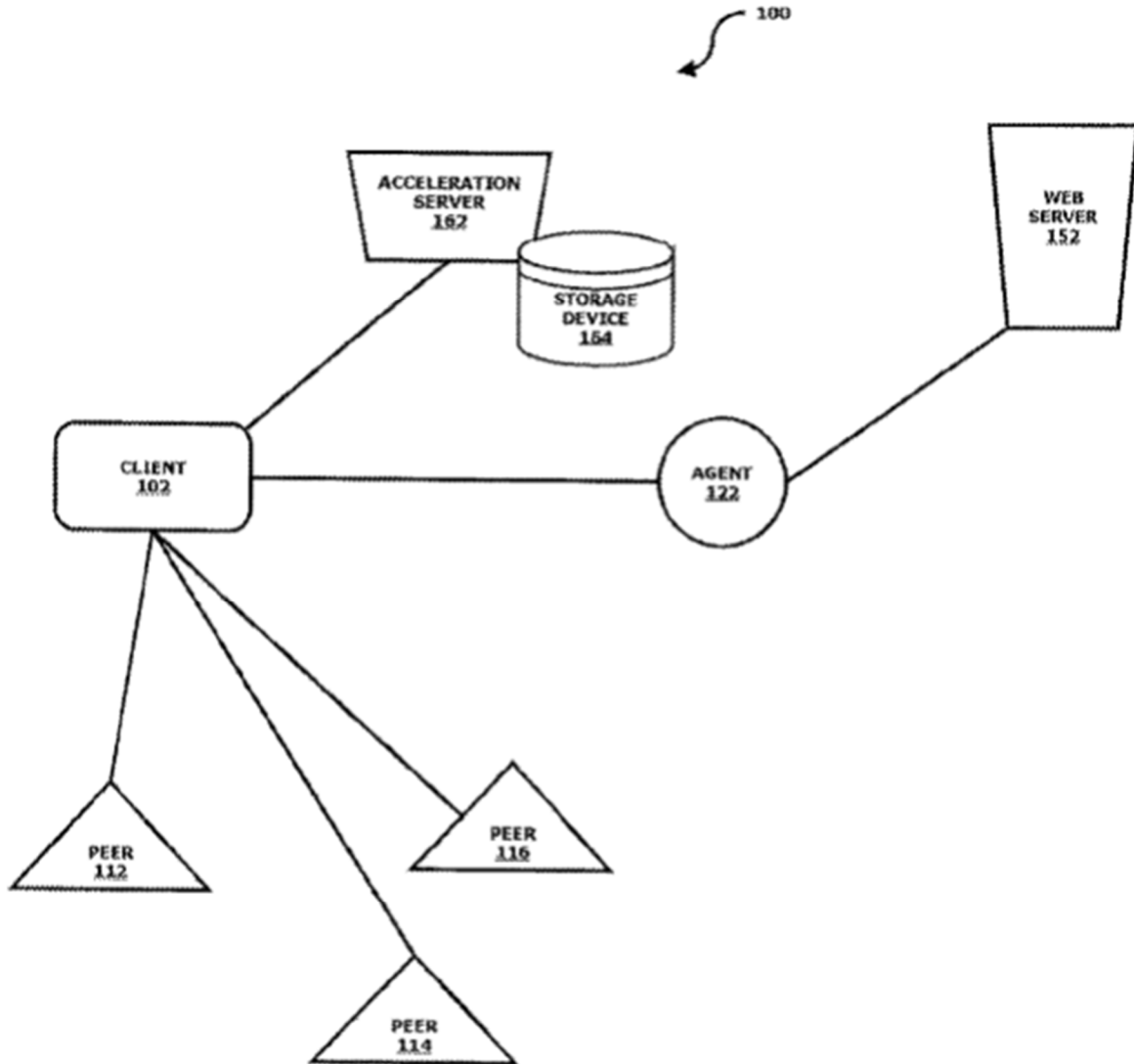


FIG. 3

Id., Fig. 3.

Client 102 sends the web request to acceleration server 162, which responds with a list of agents that may assist. *Id.*, 5:10-19, 13:8-22. Client 102 then contacts agent 122, which responds with a list of peers that have cached the requested web page, and client 102 contacts an identified peer to obtain the content. *Id.*, 13:62-64,

14:13-58, 15:14-20. The only device disclosed as contacting a web server is agent 122, which does so only when agent 122 does not identify any peers with cached content. *Id.*, 14:62-15:6. In that case, agent 122 directly contacts the web server, obtains the content, and responds to client 102 identifying itself as the only available peer—there is no “selecting” by either client 102 or agent 122 in this process. *Id.*, 15:6-11.

B. Prosecution History

The prosecution history includes no substantive analysis of any prior art; a single office action was issued raising double-patenting rejections over the related '341, '374 , and '628 Patents, which were overcome by a terminal disclaimer. EX1007, 94-98, 107-114.

C. Claim Construction (§42.104(b)(3))

The Court issued three claim construction orders in the Related Litigation construing terms of the '511, '968, '319, and '510 Patents, which have the same specification and share claim language with the Challenged Claims and are relevant to construction of the Challenged Claims.

1. “Client Device” and “First Server”

The Challenged Claims use the same “client device” and “first server” terms as the claims of the '511 Patent and Related Patents. EX1011, 5/37; EX1010, 3-4/24. Across three Orders the Court repeatedly confirmed that “client” and “server” refer to the role the device plays, not any hardware characteristics or label.

In each of the *Teso* (for the '319 and '510 Patents) and *Code200* (for the '511 and '968 Patents) Orders, the Court held that “client device” means “communication device that is *operating in the role of a client*” and rejected PO’s contention that the term required a “consumer computer.”⁸ EX1010, 10-12/24; EX1011, 11-13/37.

As to “first server,” in the *Code200* Order, the Court addressed “whether one component can simultaneously serve as more than one of: the client device, the first server[], and the web server,” concluded “[i]t cannot,” and construed “first server” as “server that is not the client device or the web server.” EX1011, 13-15/37. Subsequently, it became apparent that “server” required further construction, and Petitioner requested a clarified construction of: “a device that is operating in the role of a server and that is not the first client device or the web server.” EX1012, 2/5 (emphasis in original). The District Court declined to formally change the construction but confirmed “Defendants’ understanding of the scope of the construction[], as *represented by the requested clarification*[] ..., *is correct*” and “already embedded in” the original construction. EX1013, 7-11/12. The Court confirmed that “a component can be *configured* to operate in different roles—so long as it does not ‘simultaneously serve as more than one of: the client device, the first server/second server, and the web server.’” *Id.*, 10/12 (emphasis in original).

⁸ All emphasis added unless stated otherwise.

RFC 2616, which the '826 Patent confirms “define[s]” “the HTTP protocol,” (EX1001, 16:21-22) further confirms role-based constructions are correct: “Any given program may be capable of being both a client and a server; ***our use of these terms refers only to the role being performed*** by the program for a particular connection.” EX1005, 8/114; EX1002, ¶58.

RFC 2616 also provides further guidance on each of these “roles,” defining a “client” as “establish[ing] connections for the purpose of sending requests” and a “server” “accept[ing] connections in order to service requests by sending back responses.” EX1005, 8/114. RFC 2616 explains that a “proxy,” the component of Kocherlakota and Cohen mapped to various claim elements, is an “intermediary program which ***acts as both a server and a client*** for the purpose of making requests on behalf of other clients.” EX1005, 8/114. These RFC 2616 statements would have carried significant weight with a POSITA. EX1002, ¶59.

2. “***Server operating system***”

The '826 Specification does not use the term “server operating system” or even describe anything more than “an operating system.” *See generally* EX1001. Because the term server has been construed as ‘a device that is operating in the role of a server’, (Section VIII.C.1, above) the term server operating system should be construed as an operating system capable of operating in the role of a server. EX1002, ¶60.

Additionally, the plain and ordinary meaning of ‘server operating system’ would have been understood to be a general-purpose server operating system such as, e.g., UNIX, LINUX™, Microsoft WINDOWS NT. *See* EX1022, Col. 8:53-55 (“A general-purpose **server operating system** may be employed, e.g., UNIX, LINUX™, Microsoft WINDOWS NT®, or the like.”); EX1002, ¶61. The Board need not reach this dispute as each of Petitioner’s mappings demonstrate the art discloses either interpretation.

3. “selecting, an IP address” and “sending of the selected IP addresses to the first client device”

Limitation 1 recites “selecting, an IP address” and using “the selected IP address.” *See* Limitation 1[c], 1[d]. Limitation 19 depends directly from Limitation 1 and recites “sending of the selected IP addresses.” Thus, there is an inconsistency in the claim language between the selected IP address (singular) in claim 1 and the selected IP addresses (plural) in claim 19. Under its plain meaning, a POSITA would have understood sending *the* selected IP addresses to correspond to sending the IP address selected in claim 1. EX1002, ¶62.

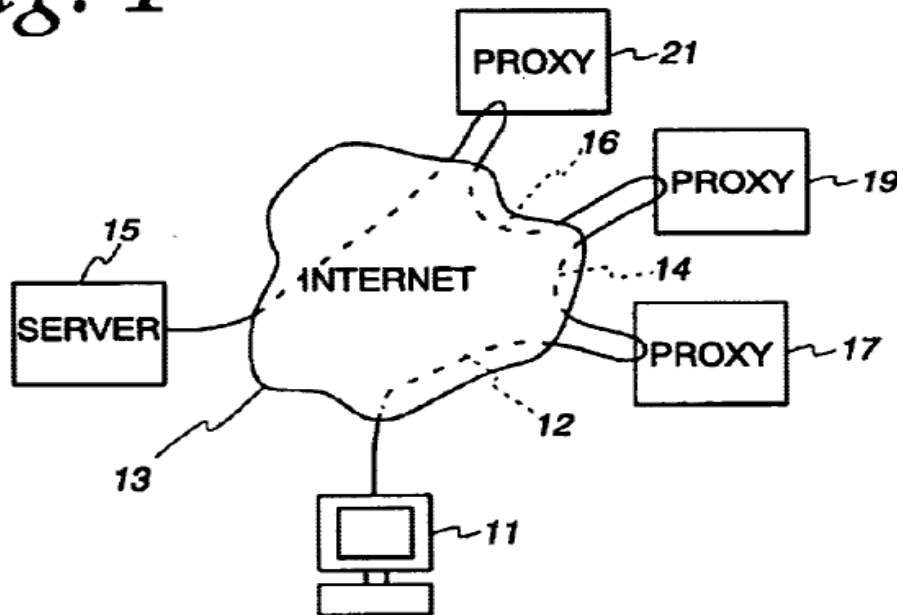
IX. REASONS FOR THE RELIEF REQUESTED (§§42.22(A)(2), 42.104(B)(4))

A. Ground 1 – Kocherlakota Renders Obvious Claims 1-5, 10, 12, 15-18, 19, and 21-25

1. Kocherlakota

Kocherlakota “relates to the use of web proxy servers by internet clients and more particularly, to the use of multiple such proxies contemporaneously.” EX1003, 1:7-9. FIG. 1 “shows an internet connection through a plurality of chained proxies.” *Id.*, 2:39-40.

Fig. 1

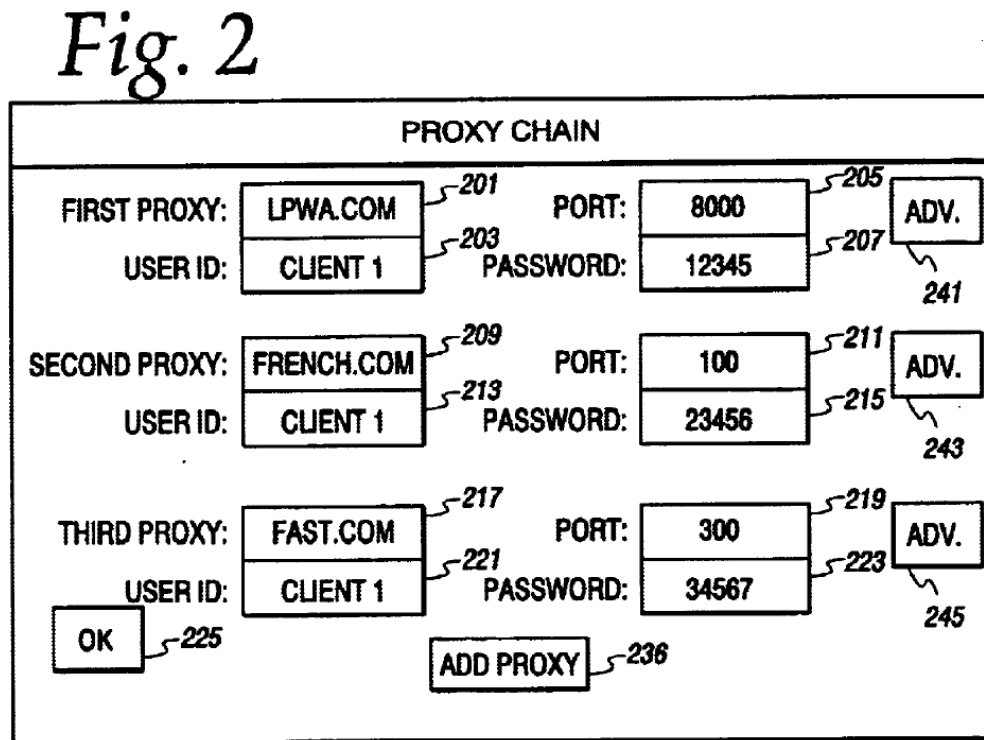


Id., FIG. 1.

In operation, client computer 11 obtains web content by sending URLs to proxy 17 for routing through a proxy chain to the web server corresponding to the URL, represented by web server 15 in FIG. 1. *Id.*, 4:29-31, 2:30-32, 2:54-61

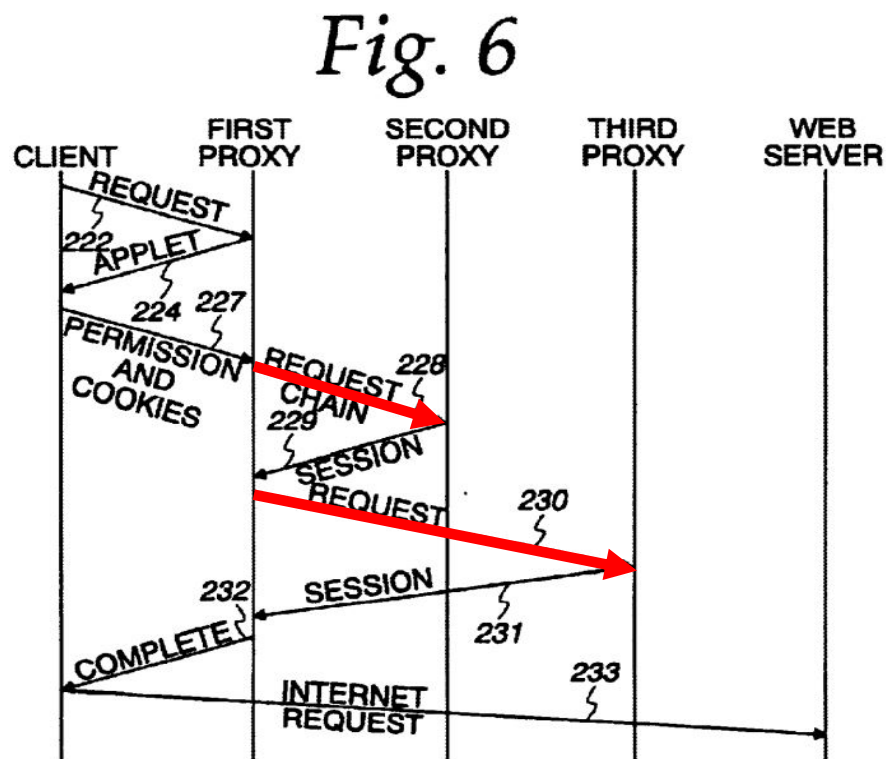
(“[W]eb server 15 ... represents the universe of addressable web servers.”); EX1002, ¶¶68-69.

Prior to client computer 11 sending URLs, proxy 17 establishes the proxy chain and routing logic for specific URLs. EX1003, 2:12-14, FIGs. 2, 7; EX1002, ¶¶71-73. Proxy 17 “transmits an applet to the client which provides a GUI interface for the client” that “presents a number of information lines into which the client can enter identities of up to three proxy servers.” EX1003, 3:29-33.



Id., FIG. 2.

As shown in FIG. 6 (annotated below), after the user identifies a second and third proxy through the applet, a request 228 is sent by the first proxy “to the second proxy which authenticates it and returns a message 229 to the first proxy signifying the session establishment.” EX1003, 4:45-48, FIG. 6; EX1002, ¶73. The first proxy then “sends a message 230 to the third proxy requesting a session...[and] the third proxy responds by returning a session message 231 to the first proxy.” EX1003, 4:48-51.

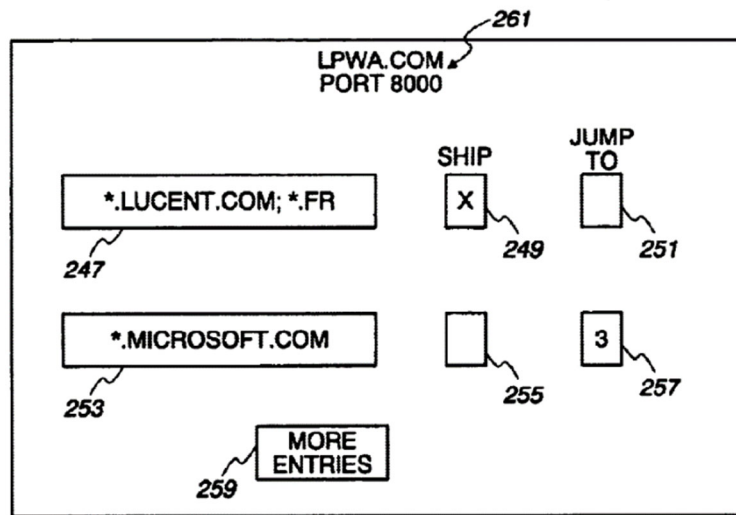


EX1003, FIG. 6.

Turning to routing through the proxy chain, “advanced features allow the user to define which proxy servers are employed for predetermined URL’s.” EX1003,

5:24-26, 4:55-5:26. That is, proxy 17 is configured as shown in FIG. 7 to perform different routing of URLs matching specific criteria. EX1002, ¶¶76-81.

Fig. 7



EX1003, FIG. 7.

First, matching criteria may be specified, as is shown in text window 247, along with a “skip” command (box 249 checked)⁹ such that “when a URL matching *.lucent.com or *.fr is transmitted to the first proxy, the services of that proxy are to be skipped as defined by the “x” in box 249.” EX1003, 5:18-21; EX1002, ¶78-79.¹⁰

⁹ Although FIG. 7 incorrectly recites “SHIP,” the specification is clear that box 249 is a “skip box.” EX1003, 5:7-21.

¹⁰ A POSITA would understand that the asterisk (*) in the matching criteria indicates that any text string may be included in place of the asterisk. EX1002, ¶73.

Second, matching criteria (e.g., *.microsoft.com) may be specified, as shown in text window 253 along with a number in a “jump to” box 257 such that “the first proxy will skip the second proxy and *jump to* the third proxy because the number 3 is entered into the jump to box” identified as 257. EX1003, 5:21-24; EX1002, ¶¶78-79. If a URL does not match the criteria in either text windows 247 or 253 (e.g., www.uspto.gov), the URL is simply “sent from the client to the internet via the chained proxies 17, 19 and 21.” EX1003, 4:29-31; EX1002, ¶¶81, 99. Thus, proxy 17 selects different proxies (e.g., proxy 19 or proxy 21) for routing based on comparing the received URL to configured rules. EX1002, ¶81.

A POSITA would understand that the session setup and maintenance, as well as subsequent routing of URLs through the proxy chains to web servers, necessarily requires, or at a minimum, renders obvious, proxy 17 storing a list of IP addresses for proxies 19 and 21 and client computer 11. EX1002, ¶82; EX1003, 4:43-54. A POSITA would understand that devices on the Internet are addressed by IP address. EX1002, ¶82 (citing EX1015, 17/51; EX1016, 57/91). Thus, in order for proxy 17 to establish and maintain sessions with proxies 19 and 21, proxy 17 would have to store the IP address corresponding to the domain name (e.g., “french.com,”

For example, www.lucent.com meets the matching criteria “*.lucent.com” and www.lucent.fr meets the matching criteria “*.fr.” *Id.*

“fast.com”) the user enters for proxies 19 and 21. *Id.*, ¶¶83.¹¹ Likewise, to establish and maintain sessions with client computer 11, proxy 17 must store the IP address of client computer 11, which would be received as the “sending address” when client computer 11 contacts proxy 17. EX1003, 3:21-22; EX1002, ¶83.

Further, a POSITA would understand that proxy 17 must also subsequently store those IP addresses so that “[a]fter the establishment of proxy sessions the client can surf the web through the established chain of proxy servers,” which necessarily requires storing for re-use the IP addresses of proxies 19 and 21 to be used for routing over the established proxy chain. EX1002, ¶84; EX1003, 2:30-32. Accordingly, when proxy 17 selects proxy 19 or 21 for routing based on the received URL, proxy

¹¹ The Challenged Claims do not recite details of how the first server obtains IP addresses to store and such processes were unknown. Kocherlakota discloses the user identifying each of proxies 19 and 21 by URL (EX1003, 3:41-48, FIG. 2), and a POSITA would understand that corresponding IP addresses must be obtained to address IP communications to each device. EX1002, ¶¶83-84. For example, using the DNS resolution processes to obtain an IP address corresponding to a URL were well-known. EX1002, ¶84; EX1001, 13:15-18; EX1005, 45, 94-95/114. Kocherlakota discloses proxy 17 “analyz[ing] the applet information” and, on that basis, “contact[ing] the named” second and third proxies 19, 21. EX1003, 4:1-20.

17 selects the IP address of the select proxy from the group of proxy IP addresses in order to route to that proxy. EX1002, ¶84.

2. Element-by-Element Analysis

Limitation 1[p]: A method for use with a web server that stores a first content identified by a first content identifier, and for use with a first client device that is addressed over the Internet using first Internet Protocol (IP) address, for use with a list of IP addresses stored in the first server, the method by the first server comprising.

Kocherlakota discloses or renders obvious this limitation. EX1002, ¶98. As discussed for the limitations of Claim 1 below, Kocherlakota discloses a method for use with web server 15 that stores a web page (the claimed “first content”) that is identified by a URL (the claimed “first content identifier”) and a client computer 11 (the claimed “first client device”) that is addressed over the Internet using an IP address (the claimed “first IP address”). EX1002, ¶98. The client computer 11 can retrieve a web page through proxy 17 (the claimed “first server”), which stores a list of IP address associated with other devices. EX1002, ¶98-99, 103.

Client computer 11 meets the construction of “first client device” because it is a device that communicates over the Internet and is operating in the role of a client by establishing a connection with proxy 17 to make requests. EX1002, ¶100; EX1003, 2:54-58 (“client computer 11 connected to the internet 13”), 4:29-31 (“the

URL of web server 15, or any other web server, can be sent from the client to the internet via the chained proxies 17, 19 and 21”); *see also* EX1003, 1:43-45, 2:4-7, 2:30-32; Section VIII.C.1.

Proxy 17 (i.e., “proxy server 17” EX1003, 4:1-31) meets the construction of “first server” because it is a device operating in the role of a server by accepting a connection from client computer 11 to service requests (URL’s) by sending back responses (content from the web server) and because it is not the same physical device as client computer 11 (the claimed “first client device”). EX1002, ¶101; EX1003, 5:3-24; Section VIII.C.1. The claimed “first content” can include a web page and the claimed “first content identifier” can include a URL. EX1002, ¶¶98, 107; *See e.g.*, EX1011, Cl. 17.

As shown in FIG. 1 of Kocherlakota, client computer 11 and proxy 17 communicate over the Internet. EX1003, FIG. 1, 3:3-8. A POSITA would understand that client computer 11 is addressed over the Internet using an IP address (the claimed “first IP address”) because a POSITA would understand that devices on the Internet are addressed by IP address. EX1002, ¶103 (citing EX1015, 17/51; EX1016, 57/91). Further, as discussed in Section IX.A.1, in order to establish sessions with proxies 19 and 21, proxy 17 must store a list of IP addresses associated with proxies 19 and 21 (the claimed “list of IP addresses”). EX1002, ¶103; EX1003, 2:30-32.

Limitation 1[a]: storing, operating, or using, a server operating system

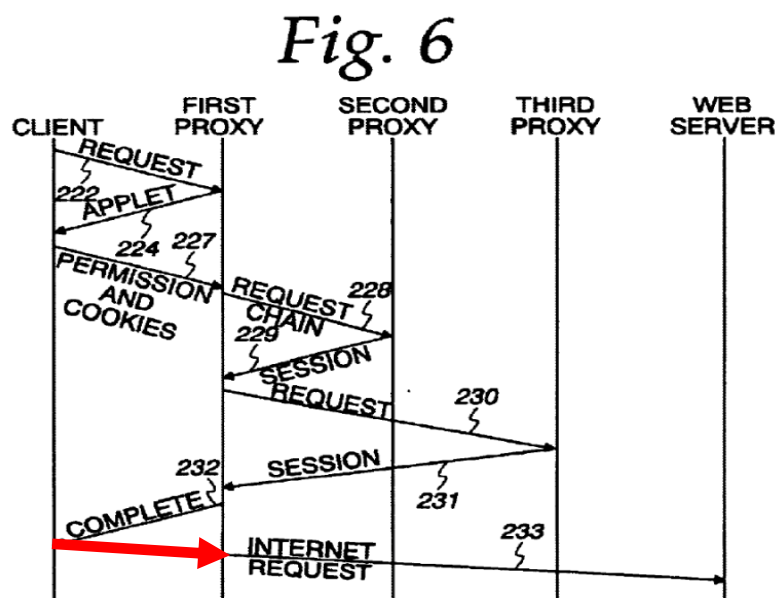
Kocherlakota discloses or renders obvious this limitation. EX1002, ¶104. Kocherlakota describes proxy 17 (the claimed “first server”) as an example of a “proxy server” that “provide services to a user ... by establishing sessions with web servers and responding to internet data from those web servers.” EX1003, 1:10-27, 4:1-31, 5:7-26.

The ’826 Patent does not use the term “server operating system” and simply recites “an operating system” as understood by its plain and ordinary meaning. EX1001, 6:31-37 (memory 210 may contain an operating system (O/S)). A POSITA would understand that computers, such as proxy 17, use an operating system and it would be, at a minimum, obvious to use a server operating system to “provide services” to clients. EX1002, ¶105 (citing EX1005, 71/114 and EX1016, 14/191).

Examples of server operating systems around October of 2009 are Unix, Linux, and Microsoft Windows NT. *See* EX1002, ¶106. It would have been obvious for a POSITA to store, operate, or use one of these operating systems in proxy 17. EX1002, ¶106.

Limitation 1[b]: receiving, from the first client device, the first content identifier

Kocherlakota discloses or renders obvious this limitation. EX1002, ¶107. As shown in FIG. 6, proxy 17 (the claimed “first server”) receives from client computer 11 (the claimed “first client device”) a URL (e.g., www.uspto.gov or www.lucent.com) as the claimed “first content identifier.”



EX1003, FIG. 6 (annotated), 2:4-7 (“the client can access the internet by sending URLs through the chained” proxies), 4:52-54; EX1002, ¶¶107-108.

As explained by the ’826 Patent, the claimed “first content identifier” includes a URL. EX1011, Cl. 17; EX1002, ¶¶98, 107. The first content identifier is a URL selected by proxy 17 by applying the logic shown in FIGs. 2 and 7. EX1002, ¶99. For example, upon receiving www.uspto.gov, proxy 17 would determine that this

URL does not match the criteria in text windows 247 or 253 of FIG. 7 and therefore route in accordance with the established proxy chain of FIG. 2, selecting proxy 19. EX1002, ¶¶99; EX1003, 4:29-31. As another example, upon receiving www.lucent.com, proxy 17 would select proxy 19 because the web page corresponding to this URL matches the criteria in text window 247 and the “skip” command 249 was selected. EX1002, ¶108; EX1003, 5:18-21.

Limitation 1[c]: selecting, an IP address from the list according to a criterion

Kocherlakota discloses or renders obvious this limitation. EX1002, ¶110. As discussed in Section IX.A.1, proxy 17 (the claimed “first server”) selects the IP address of either proxy 19 or 21 from the list of those IP addresses to route the URL through the proxy chain. EX1003, 5:10-23; EX1002, ¶111. As explained below, this selection is “according to a criterion” because the selection of the URL is based on analysis of the URL. EX1002, ¶¶112-113.

Proxy 17 performs each of the selecting steps by applying the criteria shown in FIG. 7 (i.e., applying “skip” and “jump” when a URL matches the criteria in a text window) as well as the default routing defined in FIG. 2 (each individually or collectively the claimed “criterion”). EX1003, 5:10-26. For example, when proxy 17 receives a URL not matching the criteria in either in text window 247 or 253, including www.uspto.gov, proxy 17 will not “skip” or “jump to” any server in the

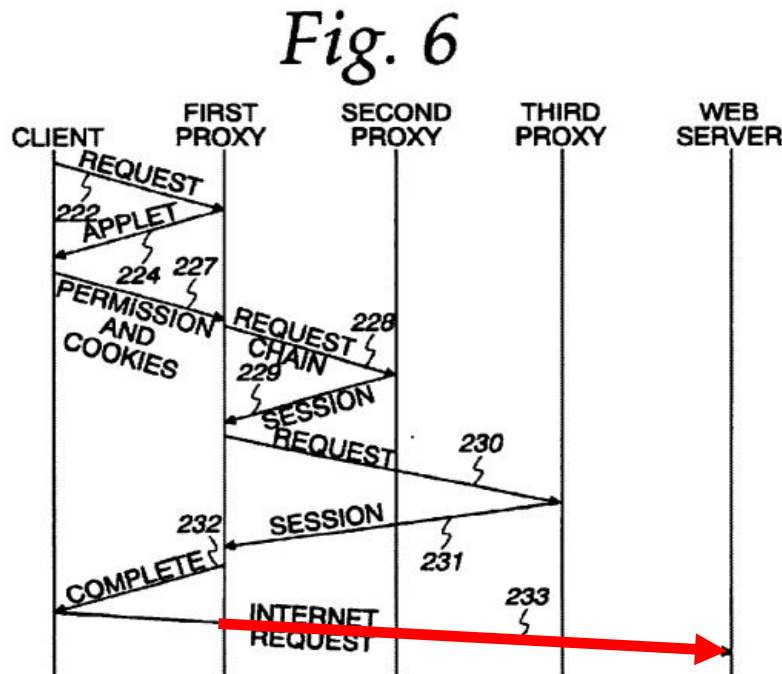
proxy chain because the URL does not match the criteria and will therefore select the IP address of the next proxy in the chain (i.e. proxy 19) so that the request passes “via the chained proxies 17, 19 and 21.” EX1002, ¶112; EX1003, 4:29-31, 5:10-23, FIGs. 6-7. Proxy 17 will thus select the IP address of proxy 19 as the address used to forward the URL. EX1002, ¶112.

Likewise, when proxy 17 receives from the client computer 11 a URL matching the criteria in text window 247 but not text window 253, including www.lucent.com, proxy 17 will “skip” the function of proxy 17, but will not “jump” over any server in the proxy chain because the URL does not match the “jump” criteria and will therefore select the IP address of proxy 19 so that the request passes “via the chained proxies 17, 19 and 21.” EX1002, ¶113; EX1003, 4:29-31, 5:10-23, FIGs. 6-7.

Limitation 1[d] sending, in response to the receiving of the first content identifier, using the selected IP address, the first content identifier

Kocherlakota discloses or renders obvious this limitation. EX1002, ¶115. As discussed in Section IX.A.1, upon receiving the URL (the claimed “first content identifier”), proxy 17 forwards the URL through the proxy chain—using the selected IP address of either proxy 19 or 21 from the list of those IP addresses. This sending of the URL is “in response to the receiving of the first content identifier.” EX1002, ¶116; Section VIII.C.2.

Proxy 17 forwards the URL to proxy 19 (in response to receiving the URL) in order for the URL to be further forwarded by proxy 19 to proxy 21 and proxy 21 to the web server hosting the first URL (the claimed “first web server”) as shown in FIG. 6:



EX1003, FIG. 6 (annotated), 2:30-32; 4:52-54; EX1002, ¶116.

A POSITA would understand that when proxy 17 sends the URL to the first web server through to proxy 19, as an intermediary, proxy 17 uses the IP address associated with proxy 19 (the claimed “selected IP address”) as the destination IP address. EX1002, ¶117.

Limitation 1[e]: receiving, in response to the sending of the first content identifier, the first content from the web server

Kocherlakota discloses or renders obvious this limitation. EX1002, ¶118. As discussed for Limitations 1[c] and 1[d], proxy 17 forwards the first content identifier (e.g., URL) along the proxy chain to be forwarded to the web server corresponding to the first content identifier. A POSITA would understand that the function of a proxy in the proxy chain is both to send URLs to web servers, as well as receive responses and return them. EX1002, ¶119. Kocherlakota discloses proxy functionality that assumes such usage. For example, Kocherlakota describes users “access[ing] the internet ‘through’ a proxy which provides value added to the ***information accessed***” and proxy servers “responding to ***internet data from those web servers***.” EX1003, 1:12-17; EX1002, ¶119. That is, client computer 11 obtains “access to web server 15 [] via the three web proxies 17, 19 and 21.” EX1003, 3:9-11. Further, proxy 19 “provide[s] translations of identified files,” which requires that proxy 19 receives the identified files from the web server. EX1003, 1:28-31, 4:29-35; EX1002, ¶119.

Kocherlakota also describes web server 15 as a “data file provider[]” and “[a]fter the establishment of proxy sessions the client can surf the web through the established chain of proxy servers.” EX1003, 2:58-61, 2:30-32. Thus, a POSITA would understand that the “web server” of Kocherlakota responds to web requests

by providing the first content (e.g., web page) to the requesting device. EX1002, ¶120.

In the disclosed proxy chain, the web server returns the web page corresponding to the first content identifier to proxy 21, proxy 21 forwards the web page to proxy 19, and proxy 19 forwards the web page to proxy 17. EX1002, ¶121; EX1003, 2:58-61, 2:30-32, 4:29-35. Proxy 17 thus receives the web page corresponding to the first URL in response to the sending of the first URL (the claimed “first content identifier”). EX1002, ¶121.

Limitation 1[f]: sending the received first content, or a part thereof, to the first client device.

Kocherlakota discloses or renders obvious this limitation. EX1002, ¶122. Client computer 11 obtains “access to web server 15 [] via the three web proxies 17, 19 and 21.” EX1003, 3:9-11. As discussed for Limitation 1[e], the function of a proxy—such as proxy 17 in the proxy chain—is both to send URLs to web servers, as well as receive responses from web servers and return them to the requesting client. EX1002, ¶122. As explained for Limitation 1[e], proxy 17 (the claimed “first server”) receives the web page corresponding to the first URL (the claimed “first content”). In accordance with its function as a proxy and providing client computer 11 “access to web server 15,” proxy 17 sends the received first content to client computer 11 (the claimed “first client device”). EX1002, ¶122.

Limitation 2: The method according to claim 1, further for use with a second client device that is addressable in the Internet using the selected IP address.

Kocherlakota discloses or renders obvious this limitation. EX1002, ¶123. Proxy 19 and proxy 21 each meet the construction of “second client device” because each is a device that communicates over the Internet and operate in the role of a client by establishing a connection down the proxy chain to make requests. EX1002, ¶123; Section VIII.C.1; *see also* EX1005, 8/114 (proxy devices “*act/s/ as* both a server and *a client*”). A session is established “between the client and a web server, e.g., 15, through proxy servers 17, 19 and 21” over which proxy 19 may make requests to proxy 21 and proxy 21 may make requests to the web server.” EX1003, 4:24-31; EX1002, ¶123.

As discussed in Section IX.A.1, to be addressable as Kocherlakota describes, a POSITA would have understood that each of the proxies 19 and 21 is identified over the Internet using a respective IP address. EX1002, ¶124; EX1015, 17/51; EX1016, 57/91. Further, as discussed for Limitation 1[c], proxies 19 and 21 are addressable over the Internet using the IP address that are selected by proxy 17 from the list of IP addresses. EX1002, ¶124.

Limitation 3: The method according to claim 2, wherein the sending of the first content identifier comprises sending to the second client device.

Kocherlakota discloses or renders obvious this limitation. EX1002, ¶125. As explained for Limitations 1[c] and 1[d], the process of proxy 17 (the claimed “first server”) sending the first URL (the claimed “first content identifier”), such as www.uspto.gov or www.lucent.com, to web server 15 includes proxy 17 sending the first URL to proxy 19 (the claimed “second client device”). EX1002, ¶125.

Limitation 4: The method according to claim 3, further comprising: receiving, by the second client device from the first server, the first content identifier; sending, by the second client device from the web server, the first content; and sending, by the second client device to the first server, the received first content or a part thereof.

Kocherlakota discloses or renders obvious this limitation. EX1002, ¶126. As explained for Limitation 3, proxy 17 (the claimed “first server”) sends the first URL (the claimed “first content identifier”) to proxy 19 (the claimed “second client device”). Thus, proxy 19 receives the first URL from proxy 17.

As explained for Limitation 1[d], the process of proxy 17 sending the first URL, e.g., www.uspto.gov or www.lucent.com to web server 15 includes proxy 19 (the claimed “second client device”) sending the first URL to proxy 21 for forwarding to the web server. EX1002, ¶127. Subsequently, as explained for

Limitation 1[e], proxy 21 receives the web page (the claimed “first content”) corresponding to the first URL from the web server and then forwards the web page to proxy 19. EX1002, ¶128; EX1003, 2:58-61, 2:30-32, 4:29-35. Thus, the web page is sent from the web server to proxy 19 and then sent, again, by proxy 19.

Web server 15 sending the web page to an intermediary, proxy 21, for eventual sending to the proxy 19 meets the claim language “sending ... from the first web server” because the web page originated from web server 15. EX1002, ¶127. Further this language (“sending ... from the first web server”) is non-limiting and includes sending to intermediaries for forwarding to the web server. EX1002, ¶127.

As further explained for Limitation 1[e], proxy 19 forwards the web page to proxy 17. EX1002, ¶128; EX1003, 2:58-61, 2:30-32, 4:29-35. Thus, the web page is sent from proxy 19 to proxy 17.

Limitation 5: The method according to claim 2, further comprising by the first server: communicating with the second client device; and storing an IP address of the second client device in the list in response to the communicating.

Kocherlakota discloses or renders obvious this limitation. EX1002, ¶129. Proxy 17 (the claimed “first server”) establishes and maintains sessions, and thereby communicates, with each of proxies 19 and 21 (the claimed “second client device”). EX1003, 4:43-54; EX1002, ¶129.

As discussed in Section IX.A.1, a POSITA would understand that the session setup and maintenance, as well as subsequent routing of URLs through the proxy chains to web servers, necessarily requires, or at a minimum renders obvious, proxy 17 storing a list of IP addresses of proxies 19 and 21. EX1002, ¶130. Where, for example, proxy 17 stores the IP addresses of proxies 19 and 21 to track an ongoing session, that storing is in response to the communicating with proxies 19 and 21 during session setup. *Id.*

Limitation 10: The method according to claim 1, wherein the criterion is stored in the first server.

Kocherlakota discloses or renders obvious this limitation. EX1002, ¶131. As discussed for Limitation 1[c], proxy 17 performs the selecting step according to the criterion shown in Kocherlakota's FIG. 7 and FIG. 2 (e.g., applying "skip" and "jump"). EX1003, 5:10-26. "The applet information and relevant session information," which include the information in FIGs. 2 and 7, is "received by the first proxy server" and, therefore, must be stored in proxy 17 to be used in routing URLs. EX1003, 3:61-67; EX1002, ¶131.

Limitation 12: The method according to claim 10, wherein the criterion is based on, or comprises, a geographical location of a client device addressed by the selected IP address, or a response time when communicating with the client device addressed by the selected IP address.

Kocherlakota discloses or renders obvious this limitation. EX1002, ¶132. Kocherlakota discloses “French language web proxy 19” provides “language service[s],” which, as discussed for Limitation 1[c], may be selected by IP address. EX1003, 2:65-67. A POSITA would understand the French language services are, or would be obvious to be, associated with a geographic location—France. EX1002, ¶132. Thus, criteria to select the IP address of proxy 19, as discussed for Limitation 14, are criteria based on or comprising geographic location. *Id.*

Further, Kocherlakota discloses matching criteria “*.fr,” which a POSITA would understand is the top-level domain for France—a geographic location. EX1003, FIG. 7; EX1002, ¶132. Kocherlakota discloses “an efficiency proxy” that “may contain a local cache for storing frequently accessed files.” EX1003, 1:31-36. It would have been obvious to have such an efficiency proxy dedicated to a specific area, such as France, which would have a unique set of “frequently accessed files,” and to use criteria for that geographic area to select that efficiency proxy based on the geographic location. EX1002, ¶132.

Additionally, each of the criteria in FIG. 7 is based on a response time because they are paired with actions—“skip[ping]” “the services of a proxy” or wholly “skip[ping]” a proxy—that impact the response time to return content to client computer 11. EX1003, 4:55-5:26; EX1002, ¶133. Further, it been obvious for a POSITA to base the selection of a proxy on response time in order to reduce the amount of time spent retrieving data. EX1002, ¶134. This selection of proxies based on geographic location and response time was a well-known and would have been a straightforward selection for a POSITA. EX1002, ¶¶135-136.

Limitation 15: The method according to claim 1, wherein the first server is a Transmission Control Protocol/Internet Protocol (TCP/IP) server that communicates over the Internet with the first client device based on, according to, or using, TCP/IP protocol or connection.

Kocherlakota discloses or renders obvious this limitation. EX1002, ¶137. As discussed for Limitations 1[p] and 1[b], proxy 17 (the claimed “first server”) is a “server” that communicates over the internet with client computer 11 (the claimed “first client device”). A POSITA would understand that all Internet protocols are based on TCP and IP. EX1002, ¶138 (citing EX1018, 1:23-27). A POSITA would also understand that the disclosed sending of URLs indicates the communication is HTTP-based. EX1003, 3:21-4:35; EX1002, ¶140. A POSITA would understand that, as RFC 2616, which the ’826 Patent recognizes “define[s]” “the HTTP

protocol,” explains, “HTTP communication usually takes place over TCP/IP connections.” EX1002, ¶140 (citing EX1001, 16:21-22, 17:20-26; EX1005, 11/114).

Further, a POSITA would understand that communicating over TCP requires establishing, by the communicating devices, a TCP session. EX1002, ¶140 (citing EX1016, 16-18/91). Thus, a POSITA would understand that Kocherlakota’s disclosure communication between proxy 17 and client computer 11 to “surf the web through the established chain of proxy servers” discloses or renders obvious proxy 17 communicating using TCP/IP, including establishing a TCP connection with client computer 11. EX1002, ¶140.

Limitation 16: The method according to claim 1, wherein the first server communicates over the Internet based on, or according to, one out of UDP, DNS, TCP, FTP, POP*, SMTP, or SQL standards.

See Limitation 15. EX1002, ¶141.

Limitation 17: The method according to claim 1, wherein the first content comprises web-page, audio, or video content, and wherein the first content identifier comprises a Uniform Resource Locator (URL).

See Limitation 1[p]. EX1002, ¶142.

Limitation 18: The method according to claim 1, for use with a plurality of client devices that includes a second client device, each identified over the Internet using a respective IP address from the list, wherein the method further comprising: selecting, by the first server, multiple client devices from the plurality of client devices by selecting the IP addresses of the multiple client devices from the list; and sending, by the first server, the selected IP addresses in response to the selecting.

Kocherlakota discloses or renders obvious this limitation. EX1002, ¶143. As explained in Limitation 2, proxy 17 is utilized with multiple other proxies, such as proxy 19 and proxy 21, each of which meet the construction of “second client device” and are addressable over the Internet using the IP address that are selected from the list of IP addresses stored in proxy 17. EX1002, ¶143.

As explained for Limitation 1[c], the process of proxy 17 selecting proxies 19 and 21 includes selecting the IP address of proxy 19 and 21 from the list of those IP addresses to route the URL through the proxy chain. EX1002, ¶144. For example, when proxy 17 receives a URL not matching the criteria in either in text window 247 or 253, proxy 17 will select the IP address of the next proxy in the chain (i.e., proxy 19). *Id*; see §IX.A.1. To further illustrate, when proxy 17 receives a URL matching the criteria of text window 253 (e.g., www.microsoft.com), proxy 17 “will skip the second proxy and jump to the third proxy because the number 3 is entered

into the jump to box 257” and will therefore select the IP address of proxy 21. Thus, proxy 17 selects the IP addresses of multiple client devices from the list of IP addresses. EX1002, ¶144.

Further, as explained for Limitation 1[d], proxy 17 sends the URL to proxy 19 in response to the selecting. Communications by proxy 17 to each of proxies 19 and 21 use the stored IP address of proxies 19 and 21 as the “destination address” in accordance with IP protocols. EX1002, ¶47 (citing EX1015, 17/51; EX1016, 57/91). When proxy 17 sends IP packets to proxy 19 or proxy 21 it uses the IP address of proxy 19 or proxy 21, respectively, (the claimed “selected IP address”) as the “destination address.” The “destination address” is part of the IP header and is thus sent with the IP packet. EX1002, ¶¶47, 117. Thus, proxy 17 sends the selected IP addresses in response to selecting the IP address. EX1002, ¶144.

Limitation 19: The method according to claim 1, further comprising sending of the selected IP addresses to the first client device.

Kocherlakota discloses or renders obvious this limitation. EX1002, ¶145. Kocherlakota teaches proxy 17 signifies (via a message) to the client 11 that the session chain is complete. EX1003 4:49-53 . (“The third proxy responds by returning a session message 231 to the first proxy and the first proxy signifies 232 to the client that the chain is complete”). It would have been obvious for proxy 17 (the claimed “first server”) to send the IP address of the proxies in the chain to client 11 (the

claimed “first client device”) in this message 232. EX1002, ¶145. Sending these IP address would have been advantageous so that the first client could verify which chain was set up and that the chain was set up correctly. *Id.* Such a modification would have represented a straightforward modification and a POSITA would have implemented with a reasonable expectation of success.

Limitation 21: The method according to claim 1, wherein the first client device is identified by a first Internet Protocol (IP) address, Media Access Control (MAC) address, or a hostname.

Kocherlakota discloses or renders obvious this limitation. EX1002, ¶146. As shown in FIG. 1, client computer 11 (the claimed “first client device”) and proxy 17 (the claimed “first server”) communicate over the Internet. EX1003, FIG. 1, 3:3-8. A POSITA would understand that client computer 11 is addressed over the Internet using an IP address (the claimed “first IP address”) because a POSITA would understand that devices on the Internet are addressed by IP address. EX1002, ¶146 (citing EX1015, 17/51; EX1016, 57/91). Thus, client computer is identified by an IP address. EX1002, ¶146.

Limitation 22: The method according to claim 1, wherein the web server is a Hypertext Transfer Protocol (HTTP) server responding to HTTP requests.

Kocherlakota discloses or renders obvious this limitation. EX1002, ¶147. Kocherlakota describes “web server 15” (the claimed “web server”) as

“representative of any of the well-known data file providers and, in fact, represents the universe of addressable web servers.” EX1003, 2:58-61. HTTP servers responding to HTTP requests were “well known data file providers.” EX1002, ¶147 (citing EX1019, 2:43-53; EX1005, 94/114); EX1001, 4:63-67.

Limitation 23: The method according to claim 1, wherein the steps are sequentially executed.

Kocherlakota discloses or renders obvious this limitation. EX1002, ¶148. As discussed for Limitations 1[p]-1[f], Kocherlakota discloses proxy 17 performing the steps of Claim 1 in a sequential manner. *Id.* The claimed steps follow the known function of a proxy—sending URLs to web servers, receiving responses from web servers, and returning them to the requesting client. *Id.*

Limitation 24: The method according to claim 1, for use with a software application that includes computer instructions that, when executed by a computer processor, cause the processor to perform the steps of the claim 1.

Kocherlakota discloses or renders obvious this limitation. EX1002, ¶149. As discussed for Limitations 1[p]-1[f], Kocherlakota discloses proxy 17 performing the steps of Claim 1. A POSITA would understand that, as is the nature of computing devices, proxy 17 would include a processor and perform those steps by executing a software application that includes computer instructions causing the processor to perform the steps. EX1002, ¶149.

Limitation 25: A non-transitory computer readable medium containing computer instructions that, when executed by a computer processor, cause the processor to perform the method according to claim 1.

Kocherlakota discloses or renders obvious this limitation. EX1002, ¶150. As discussed for Limitations 1[p]-1[f], and limitation [25] Kocherlakota discloses proxy 17 performing the steps of Claim 1. A POSITA would understand that, as is the nature of computing devices, proxy 17 would include a processor and a non-transitory computer readable medium containing computer instructions that are to be executed by the processor to cause the processor to perform the steps. EX1002, ¶150.

B. Ground 2 – Kocherlakota in View of RFC 1122 Renders Obvious Claims 6-9,

With Ground 1 as predicate, Challenged Claims 6-9 would have been obvious to a POSITA based upon Kocherlakota in view of RFC 1122.

1. RFC 1122 Overview

Request for Comments (“RFC”) 1122 is titled “Requirements for Internet Hosts – Communication Layers” and provides requirements for link layer, Internet layer, and transport protocols. EX1002, ¶86. It includes a section describing various features of TCP, including the use of a “‘keep-alive’ mechanism” “[t]o confirm that an idle connection is still active.” EX1006, 102/116; EX1002, ¶88. “A ‘keep-alive’ mechanism periodically probes the other end of a connection when the connection

is otherwise idle, even when there is no data to be sent.” EX1006, 102/116. “To confirm that an idle connection is still active,” some TCP “implementations send a probe segment designed to elicit a response from the peer TCP.” *Id.* “[T]he probe causes the receiver to return an acknowledgment segment, confirming that the connection is still live. If the peer has dropped the connection due to a network partition or a crash, it will respond with a RST instead of an acknowledgment segment.” *Id.*

2. Motivation to Combine

A POSITA would have been motivated to use the TCP keep-alive functionality of RFC 1122 when maintaining TCP sessions as disclosed by Kocherlakota. EX1002, ¶152. Both references cover TCP connections, and a POSITA would have recognized the benefits of Kocherlakota’s proxy servers exchanging TCP keep-alives with each other. EX1002, ¶152-153. As one benefit, TCP keep-alives help detect when “a client crashes or aborts a connection during a network failure” and thus prevent server applications from “hang[ing] indefinitely and consum[ing] resources unnecessarily.” EX1006, 102/116; EX1002, ¶153. For example, following the establishment of sessions between proxy 17 and the client computer 11, proxy 19 and/or proxy 21, as discussed in Sections IX.A.1 and IX.B.1, one of client computer 11, proxy 19, or proxy 21 could crash. EX1002, ¶153. Without TCP keep-alives, client computer 11, proxy 19, or proxy 21 would

unnecessarily expend resources keeping the TCP connection open. EX1002, ¶153. As another independent example, a POSITA would have known that TCP keep-alives could also prevent periods of inactivity between client devices from causing the TCP connection between them to terminate. EX1002, ¶153.

RFC 1122 does specify some potential disadvantages of TCP keep-alives, but a POSITA would have recognized that these disadvantages are outweighed by the benefits listed above. EX1002, ¶154; *see Winner Int’l Royalty Co. v. Wang*, 202 F.3d 1340, 1349 incl. n. 8 (Fed. Cir. 2000) (To assess the obviousness of a combination, “the benefits, both lost and gained, should be weighed against one another,” and “[m]otivation to combine requires” “what is, on balance, desirable.”); *Allied Erecting & Dismantling Co. v. Genesis Attachments, LLC*, 825 F.3d 1373, 1381 (Fed. Cir. 2016) (“given course of action often has simultaneous advantages and disadvantages, and this does not necessarily obviate motivation to combine.”).

3. *Modification of Kocherlakota in view of RFC 1122*

A POSITA would have implemented the system of Kocherlakota such that during sessions between proxy 17 and client computer 11, proxy 19 and/or proxy 21, the devices would periodically communicate by exchanging TCP keep-alive messages with each other over established TCP connections. EX1002, ¶155. As explained in Sections IX.B.1 and IX.B.2, a POSITA would recognize that the “connection” RFC 1122 refers to is a TCP connection. In the proxy chain

comprising a client computer 11 -> proxy 17 -> proxy 19 or the proxy chain comprising a computer 11 -> proxy 17 -> proxy 21, the proxy 17 would periodically send client computer 11 and proxies 19 and/or 21 keep-alive messages over the TCP connections between them. *Id.* Further, in accordance with RFC 1122, the proxy 17 would wait for an “acknowledgment” from proxies 19 and/or 21, confirming that the connection is still live. EX1006, 102/116; EX1002, ¶155.

4. Element-by-Element Analysis

Limitation 6: The method according to claim 1, for use with a second client device, the method further comprising: sending, by the first server, a ‘keep alive’ message to the second client device; and waiting, by the first server, for a response to the ‘keep alive’ message from the second client device.

Kocherlakota-RFC 1122 renders obvious this limitation. EX1002, ¶157. As discussed for Limitation [2], proxies 19 and 21 meet the construction of “second client device.” As discussed in Section IX.B.3, proxy 17 (the claimed “first server”) sends keep-alive messages to proxies 19 and 21 (the claimed “second client devices”) and waits for response from each. EX1002, ¶¶157-158.

Limitation 7: The method according to claim 6, further comprising: receiving, by the first server, the response to the ‘keep alive’ message from the second client device; and responsive to the receiving of the response, adding the IP address of the second client device to the list.

Kocherlakota-RFC 1122 renders obvious this limitation. EX1002, ¶159. As explained in Section IX.B.3 and Limitation 6, proxy 17 (the claimed “first server”) waits for an “acknowledgment” from proxies 19 and/or 21, confirming that the connection is still live. If the proxies 19 and/or 21 (the claimed “second client device”) are live, then proxy 17 will receive the acknowledgment (the claimed “response to the ‘keep alive’ message”) and determine the connection is available. EX1002, ¶159.

The addition of TCP keep-alives from RFC1122 would have been very simple for a POSITA to set up and use. EX1002, ¶159. For example, as described in ‘TCP Keepalive HOWTO,’ if you receive a reply to the probe packet the connection is assumed to be alive. EX1021, §2.1. However, if you do not receive a reply from the probe packet, a count is kept of the unacknowledged probes; and once this count equals a set value (“tcp_keepalive_probes”), the connection is considered dead. EX1021, §3. Thus, there exists a state when the connection is neither alive or dead (i.e., a reply is not received and the count is less than the set value). EX1002, ¶159. In this unidentified state it would have been prudent for the POSITA to remove this

proxy from the list of available proxies, as opposed to leaving it on the available proxy list in case the proxy itself was dead. EX1002, ¶159. However, once a keep-alive probe is subsequently received by proxy 17 a POSITA would place the device from which the tcp-keepalive probe was received back on the list of available proxies. EX1002, ¶159. Such a modification would have been straightforward and merely represents combining prior art elements according to known methods to yield predictable results. Additionally, the combination is the use of a known technique to improve similar devices in the same way.

Thus, as explained above and for Limitation 1, proxy 17 would have added the IP address of proxies 19 and/or 21 to the list in response to receiving the keep-alive probe reply.

Limitation 8: The method according to claim 6, further comprising responsive to not receiving, by the first server, a response to the ‘keep alive’ message from the second client device, removing the IP address of the second client device from the list.

Kocherlakota-RFC 1122 renders obvious this limitation. EX1002, ¶161. As discussed for Limitation 7, it would have been obvious to remove the IP address of a non-responsive client device from the list of IP addresses of devices under active session. EX1002, ¶161. If proxy 17 determines a connection is not active, then it may consider the session terminated and remove the IP address of the terminated

device from the list of IP addresses of devices under active session. *Id.* Thus, proxy 17 maintains a list of IP addresses whose respective client devices are current only. *Id.*

Limitation 9: The method according to claim 6, wherein the sending of the ‘keep alive’ message comprises periodically sending by the first server the ‘keep alive’ message to the second client device.

Kocherlakota-RFC 1122 renders obvious this limitation. EX1002, ¶162. As discussed in Section IX.B.3, proxy 17 (the claimed “first server”) periodically sends keep-alive messages to proxies 19 and 21 (the claimed “second client device”). EX1002, ¶162.

C. Ground 3– Cohen Renders Obvious Claims 1-6, 12, 15-17, 18, 19, and 21-25

1. Cohen

Cohen discloses a system for transparently intercepting client web requests and redirecting them to proxy caches. EX1004, 1:7-9; EX1002, ¶91. “Proxy caching is currently used to decrease both the latency of object retrieval and traffic on the Internet backbone.” *Id.*, 1:13-18.

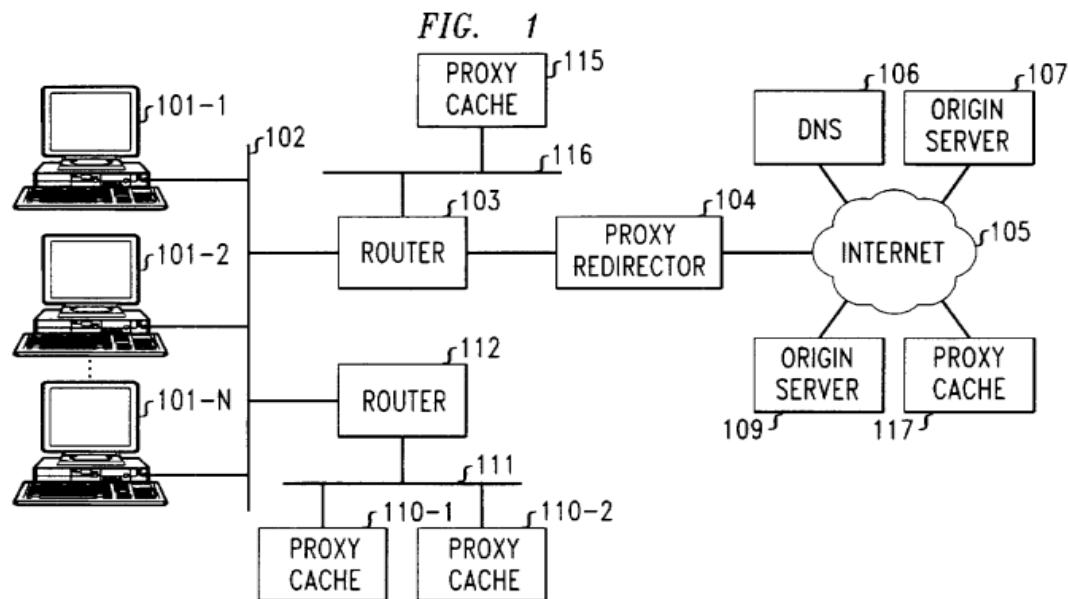
The process described in Cohen begins when a client makes a request with a URL through a web browser, which initiates a DNS lookup:

When any of the clients connected to LAN 102, such as client 101-1, makes a request through a browser for an object by specifying a logical

URL, a domain name server (DNS) 106 connected locally or on Internet 105, as shown, is accessed to perform a database look-up based on that logical name. An associated IP address is then returned to the browser.

EX1004, 6:47-65. The “IP address then serves as the IP address to which packets directed to the origin server from the client are directed.” *Id.*; EX1002, ¶92.

Then, “once the IP address of the origin server corresponding to the logical URL name is determined through the DNS look-up, proxy redirector 104, rather than establishing a TCP connection to the origin server at the determined IP address, transparently establishes a TCP connection between the client and a proxy.” EX1004, 7:12-19; FIG. 1 (reproduced below).



Proxy redirector 104 intercepts a web request from a “client 101-1” seeking content from a target web server and “selects one of the available proxy caches [e.g.

110-1, 110-2, 115, and/or 117] to which to forward client requests based on a metric such as least-loaded or round-robin, [or] based on IP header information such as the origin server IP address.” *Id.*, 7:36-48; *see also id.*, 2:15-20 (client sends “one or more packets to the origin server [via the proxy redirector 104], which packets include a GET request. The GET request includes a complete URL, which identifies to that server the specific object within the origin server site that the client wants a copy of.”). Cohen further explains that stored IP addresses are used to identify the selected proxy. *Id.*, 8:29-31 (proxy redirector program “translates the destination IP address of packets addressed to the origin server to the IP address of a selected proxy cache”); *Id.*, 9:3 (“...store[s]...chosen proxy’s IP address”); *Id.*, 9:6-8 (“All subsequent packets that originate from the same client with the same TCP port number are then forwarded to the same proxy.”).

Proxy redirector 104 then forms the absolute URL by prefixing the complete URL “with the IP address of the origin server obtained from the destination IP address of the packet(s) containing the GET request.” EX1004, 9:28-10:3. “If the proxy cache to which the request is directed does not contain the requested object, a separate TCP connection is established between the proxy cache and the origin server to obtain a copy of the requested object.” *Id.*, 7:27-35. Finally, “responsive packets from the proxy cache are sent to proxy redirector 104 where they are redirected to the client.” EX1004, 8:55-58.

2. *Element-by-Element Analysis*

Limitation 1[p].¹²

Cohen discloses or renders obvious this limitation. EX1002, ¶163. As discussed in Section IX.C.1 and for the limitations of Claim 1 below, Cohen discloses a method for use with an origin server that stores a web page (the claimed “first content”) identified by a first URL (the claimed “first content identifier) and a client 101-1 (the claimed “first client device”) that is addressed over the Internet using an IP address (the claimed “first IP address”). EX1002, ¶163. The client 101-1 can retrieve a web page through proxy redirector 104 (the claimed “first server”), which stores a list of IP address associated with other devices. EX1002, ¶164-167.

Client 101-1 meets the construction of “first client device” because it is a device that communicates over a network, and it is operating in the role of a client by making requests to proxy redirector 104. EX1002, ¶168; EX1004, 8:55-58 (“...packets from a client 101-1 are transparently directed by proxy redirector 104 to a proxy cache. Responsive packets from the proxy cache are sent to proxy redirector 104 where they are redirected to client 101-1.”).

¹² The text of each limitation in Ground 3 is included above and in-full in either Ground 1 or Ground 2, as well as in the Claim Listing.

Proxy redirector 104 meets the construction of “first server” because it is a device operating in the role of a server by receiving client 101-1’s web requests and sending the responses back to client 101-1 and because it is not the same physical device as client 101-1 (the claimed “first client device”). EX1002, ¶169; EX1004, 8:55-58. As explained in Section IX.C.1, in communicating with proxy redirector 104, the client 101-1 sends a GET request which includes a complete URL (the claimed “first content identifier”).

Proxy redirector 104 receives a web request from a “client 101-1” seeking content from a target web server and “selects one of the available proxy caches to which to forward client requests based on a metric such as least-loaded or round-robin, [or] based on IP header information such as the origin server IP address.” EX1004, 7:36-48; EX1002, ¶167. Further, as discussed in Section IX.C.1, in order to select one of the many available proxy caches, proxy redirector 104 must store a list of IP addresses associated with each proxy cache (the claimed “list of IP addresses”). EX1002, ¶167; EX1004, 7:43-48; 8:29-31; 9:3. As Cohen explains, IP addresses are used to identify proxies, so the proxy redirector can choose an available proxy cache. Proxy redirector 104 “translates the destination IP address of packets addressed to the origin server to the IP address of a selected proxy cache,” which requires proxy redirector 104 to have stored that IP address. EX1004, 7:43-48; 8:29-31; 9:3. EX1002, ¶¶167.

Limitation 1[a]:

Cohen discloses or renders obvious this limitation. EX1002, ¶170. Cohen discloses that the proxy redirector (“first server”) may be a Level 4 switch that also may be combined into the same unit as a router. EX1004, 6:23-31. The functions of any switches “may be carried out through the operation of program logic, through dedicated logic, through the interaction of program control and dedicated logic, or even manually, the particular technique being selectable by the implementor as more specifically understood from the context.” *Id.*, 17:58-63. The computer functions may be carried out by “software in any form, including, therefore, firmware, microcode or the like, combined with appropriate circuitry for executing that software to perform the function.” *Id.*, 18:1-3. Cohen further discloses that the “[i]n the preferred embodiment described herein, proxy redirector 104 includes a programmable network element ... that programmable network element in the preferred embodiment runs on a Linux machine.” *Id.*, 7:49-55. FIG. 2 of Cohen illustrates this programmable network element and that it runs on a “Linux Kernel.” EX1004, Fig 2. A POSITA would understand Linux to be a well-known server operating system. EX1002, ¶170.

Cohen further discloses that “proxy redirector 104 includes a programmable network element of the type described in co-pending U.S. patent application Ser. No. 09/190,355, filed Nov. 12, 1998, which application is incorporated herein by

reference. As described in that application, that programmable network element in the preferred embodiment runs on a Linux machine.” *Id.*, 7:49-55. Accordingly, a POSITA would understand this to intrinsically disclose a server operating system, which would have been well known to a POSITA. EX1002, ¶170.

Limitation 1[b]:

Cohen discloses or renders obvious this limitation. EX1002, ¶171. Cohen discloses that the client sends “one or more packets to the origin server [via the proxy redirector 104], which packets include a GET request. The GET request includes a complete URL, which identifies to that server the specific object within the origin server site that the client wants a copy of.” EX1004, 2:15-20.

As shown in Table 1 (annotated below), the proxy redirector 104 (the claimed “first server”) receives the GET request from client 101-1 (the claimed “first client device”) with the included complete URL “/a/ya/yahoomail/promo1.gif HTTP/1.0” (the claimed “first content identifier”). EX1004, 9:19-23; Table 1; EX1002, ¶172; *see also* EX1004, 2:15-20. Accordingly, the proxy redirector 104 receives the web request and URL from the client 101-1. As discussed for Limitation 17, the claimed “first content identifier” includes a URL. EX1002, ¶¶163, 198.

TABLE 1

Arriving packet:

```

---> beginning of packet header dump <---
---> IP header: version=4 hdr_len=5 TOS=0 pkt_len=346 id=60276
      frag_off=4000H TTL=64 protocol=6 cksum=1792H
      saddr=135.104.25.243 daddr=204.71.200.244
---> TCP header: sport=1273 dport=80 seq=2189084427 ack_seq=3266449517
      tcp_hdr_len=5 flags=ACK PSH
      res1=0H res2=0H window=31856 cksum=162H urgent=0

```

---> beginning of packet data dump <---

GET /a/ya/yahoomail/promo1.gif HTTP/1.0

Referer: http://www.yahoo.com/

Connection: Keep-Alive

User-Agent: Mozilla/4.05 [en] (X11; U; Linux 2.1.103 i686)

Host: us.yimg.com

Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg image/png

Accept-Language: en

Accept-Charset: iso-8859-1, *,utf 8

Modified to:

```

---> beginning of packet header dump <---
---> IP header: version=4 hdr_len=5 TOS=0 pkt_len=367 id=60276
      frag_off=4000H TTL=64 protocol=6 cksum=1792H
      saddr=135.104.25.245 daddr=135.104.25.31
---> TCP header: sport=5000 dport=7000 seq=2189084427 ack_seq=3266449517
      tcp_hdr_len=5 flags=ACK PSH
      res1=0H res2=0H window=31856 cksum=162H urgent=0

```

---> beginning of packet data dump <---

GET http://204.71.200.244/a/ya/yahoomail/promo1.gif HTTP/1.0

Referer: http://www.yahoo.com/

Connection: Keep-Alive

User-Agent: Mozilla/4.05 [en] (X11; U; Linux 2.1.103 i686)

Host: us.yimg.com

Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg image/png

Accept-Language: en

Accept-Charset: iso-8859-1, *,utf 8

Limitation 1[c]:

Cohen discloses or renders obvious this limitation. EX1002, ¶173. As discussed in Section IX.C.1 and as shown in Table 1 (annotated below), proxy redirector 104 (the claimed “first server”) selects the IP address “daddr=135.104.25.31” of a proxy cache 110-1 from the list of IP addresses

associated with at least two or more proxies (i.e. proxies 110-1, 110-2, 115, and/or 117) to forward the URL to the selected proxy. EX1004, 11:12-15; EX1002, ¶173.

TABLE 1

Arriving packet:

```

---> beginning of packet header dump <---
---> IP header: version=4 hdr_len=5 TOS=0 pkt_len=346 id=60276
      frag_off=4000H TTL=64 protocol=6 cksum=1792H
      saddr=135.104.25.243 daddr=204.71.200.244
---> TCP header: sport=1273 dport=80 seq=2189084427 ack_seq=3266449517
      tcp_hdr_len=5 flags=ACK PSH
      res1=0H res2=0H window=31856 cksum=162H urgent=0
---> beginning of packet data dump <---
GET /a/ya/yahoomail/promo1.gif HTTP/1.0
Referer: http://www.yahoo.com/
Connection: Keep-Alive
User-Agent: Mozilla/4.05 [en] (X11; U; Linux 2.1.103 i686)
Host: us.yimg.com
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg image/png
Accept-Language: en
Accept-Charset: iso-8859-1, *,utf 8
Modified to:

---> beginning of packet header dump <---
---> IP header: version=4 hdr_len=5 TOS=0 pkt_len=367 id=60276
      frag_off=4000H TTL=64 protocol=6 cksum=1792H
      saddr=135.104.25.245 daddr=135.104.25.31
---> TCP header: sport=5000 dport=7000 seq=2189084427 ack_seq=3266449517
      tcp_hdr_len=5 flags=ACK PSH
      res1=0H res2=0H window=31856 cksum=162H urgent=0
---> beginning of packet data dump <---
GET http://204.71.200.244/a/ya/yahoomail/promo1.gif HTTP/1.0
Referer: http://www.yahoo.com/
Connection: Keep-Alive
User-Agent: Mozilla/4.05 [en] (X11; U; Linux 2.1.103 i686)
Host: us.yimg.com
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg image/png
Accept-Language: en
Accept-Charset: iso-8859-1, *,utf 8

```

As explained in Section IX.C.1, proxy redirector 104 uses a stored criterion when it “selects one of the available proxy caches.” EX1002, ¶174. For example, “[p]roxy redirector 104 selects one of the available proxy caches ... based on “a

metric such as least-loaded or round-robin [or] IP header information such as the origin server IP address.” EX1004, 7:43-47. With respect to the latter, all objects from a specific origin server will be served by a specific proxy.” EX1004, 7:36-48.

Limitation 1[d]:

Cohen discloses or renders obvious this limitation. EX1002, ¶175. As discussed for Limitation 1[c], the proxy redirector 104 selects the IP address of proxy cache 110-1 and forwards the URL to proxy cache 110-1. EX1004, 7:36-48; 11:12-15 (“As can be noted in the modified packet, the packet is redirected to a proxy cache 110-1 at IP address 135.104.25.31 port 7000 by changing the daddr and dport header information.”); EX1002, ¶175. This sending of the URL is “in response to the receiving of the first content identifier” because the selection and sending of the URL is “based on IP header information such as the origin server IP address.” EX1002, ¶177. EX1004, 7:43-48. This origin server IP address corresponds to the URL and, thus, reflects analysis of the URL is “selecting.” EX1002, ¶177.

A POSITA would understand that when proxy redirector sends the URL to the first web server through to proxy cache 110-1, as an intermediary, proxy redirector uses the IP address associated with proxy cache 110-1 (the claimed “selected IP address”) as the destination IP address. EX1002, ¶176.

Limitation 1[e]:

Cohen discloses or renders obvious this limitation. EX1002, ¶178. “[P]ackets from a client 101-1 are transparently directed by proxy redirector 104 to a proxy cache. Responsive packets from the proxy cache are sent to proxy redirector 104 where they are redirected to client 101-1.” EX1004, 8:55-58; *see id.*, 6:39-46, 10:23-25, 12:18-20. Cohen further explains that the proxy cache returns the “requested objects” (i.e., the claimed “first content”) to the address from which a request originated. EX1002, ¶179; EX1004, 16:28-41.

Limitation 1[f]:

Cohen discloses or renders obvious this limitation. EX1002, ¶180. As explained above for Limitation 1[e], “[r]esponsive packets from the proxy cache are sent to proxy redirector 104 where they are redirected to client 101-1.” EX1004, 8:55-58. Thus, the proxy redirector 104 returns the “requested objects” (i.e., the claimed “first content”) to client 101-1. EX1002, ¶180; EX1004, 16:28-41.

Limitation 2:

Cohen discloses or renders obvious this limitation. EX1002, ¶181. Each of the proxies 110-1, 110-2, 115, and/or 117 each meet the construction of “second client device” because each is a device that communicates over the Internet and operate in the role of a client by establishing connections between the origin server and the proxy redirector 104. EX1002, ¶181; Section IX.C.1 and Limitation 1[c];

EX1005, 8/114 (proxy devices “*act as* both a server and *a client*”). As discussed in Section IX.C.1, proxy redirector 104 identifies proxies 110-1, 110-2, 115, and/or 117 using a respective IP address. EX1002, ¶182. Further, as discussed for Limitation 1[c], the proxies are addressable over the Internet using the IP address that are selected by proxy redirector 104 from the list of IP addresses.

Limitation 3:

Cohen discloses or renders obvious this limitation. EX1002, ¶183. As explained in Section IX.C.1 and Limitations 1[c] and 1[d], the process of proxy redirector 104 (the claimed “first server”) sending the GET request (containing the URL, e.g. “/a/ya/yahoomail/promo1.gif HTTP/1.0”) to the origin server (e.g. origin server 107 or 109) includes proxy redirector 104 sending the GET request and URL (the claimed “first content identifier”) to proxy cache 110-1 (the claimed “second client device”). EX1002, ¶183.

Limitation 4:

Cohen discloses or renders obvious this limitation. EX1002, ¶184. As explained for Limitation 3, proxy redirector 104 sends the GET request with the URL to proxy cache 110-1. *Id.* Thus, proxy cache 110-1 (the claimed “second client device”) receives the GET request with the URL (the claimed “first content identifier”) from proxy redirector 104 (the claimed “first server”). *Id.*

As explained for Limitation 1[d], the process of proxy redirector 104 sending the GET request containing the URL to the origin server includes proxy cache 110-1 sending the URL to the web server through a TCP connection that “is established between the proxy cache and the origin server to obtain a copy of the requested object.” EX1004, 7:27-35. EX1002, ¶185. As explained for Limitation 1[e], proxy cache 110-1 receives from the origin server the web page corresponding to the URL and forwards the web page to proxy redirector 104. EX1002, ¶186; EX1004, 8:55-58, 6:39-46, 10:23-25, 12:18-20. Thus, the web page is sent from the origin server (the claimed “web server”) to proxy cache 110-1 and then is sent, again, by proxy cache 110-1 to proxy redirector 104. EX1002, ¶187.

Limitation 5:

Cohen discloses or renders obvious this limitation. EX1002, ¶¶188, 189. Proxy redirector 104 (the claimed “first server”) establishes and maintains connections to determine availability, and thereby communicates, with each of the proxy caches 110-1, 110-2, 115, and/or 117, (the claimed “second client devices”). EX1004, 7:36-48; EX1002, ¶188.

As discussed in Section IX.C.1, a POSITA would understand that when proxy redirector 104 “selects one of the available proxy caches [e.g. 110-1, 110-2, 115, and/or 117] to which to forward client requests, ” such connection necessarily requires, or at a minimum renders obvious, proxy redirector 104 storing a group of

IP addresses for the proxies. EX1002, ¶189; EX1004, 7:36-48, 2:15-20. Further, a POSITA would also understand that when proxy redirector 104 “transparently establishes a TCP connection between the client and a proxy” such connection necessarily requires, or at a minimum, renders obvious, proxy redirector 104 storing the IP address of the proxy. EX1002, ¶189; EX1004, 7:12-17.

Limitation 6:

Cohen discloses or renders obvious this limitation. EX1002, ¶¶190. As discussed for Limitation [2], proxies 110-1, 110-2, 115, and/or 117 meet the construction of “second client device.”

As discussed in Section IX.C.1 and as shown in Table 1 (annotated below), proxy redirector 104 establishes a “keep alive” connection with the client (e.g. 101-1) and the proxies. EX1004, 11:12-15; EX1002, ¶190. This HTTP “keep-alive” feature allows the HTTP client (i.e., the client 101-1 or cache proxy) to send multiple request/response pairs over the same TCP connection. *Id.* This decreases latency for subsequent HTTP requests since the proxy redirector 104 (the claimed “first server”) does not close the TCP connection after each response but waits some time to allow for additional HTTP requests. *Id.*

TABLE 1

Arriving packet:

```

---> beginning of packet header dump <---
---> IP header: version=4 hdr_len=5 TOS=0 pkt_len=346 id=60276
      frag_off=4000H TTL=64 protocol=6 cksum=1792H
      saddr=135.104.25.243 daddr=204.71.200.244
---> TCP header: sport=1273 dport=80 seq=2189084427 ack_seq=3266449517
      tcp_hdr_len=5 flags=ACK PSH
      res1=0H res2=0H window=31856 cksum=162H urgent=0
---> beginning of packet data dump <---
GET /a/ya/yahoomail/promo1.gif HTTP/1.0
Referer: http://www.yahoo.com/
Connection: Keep-Alive
User-Agent: Mozilla/4.05 [en] (X11; U; Linux 2.1.103 i686)
Host: us.yimg.com
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg image/png
Accept-Language: en
Accept-Charset: iso-8859-1, *,utf 8
Modified to:

---> beginning of packet header dump <---
---> IP header: version=4 hdr_len=5 TOS=0 pkt_len=367 id=60276
      frag_off=4000H TTL=64 protocol=6 cksum=1792H
      saddr=135.104.25.245 daddr=135.104.25.31
---> TCP header: sport=5000 dport=7000 seq=2189084427 ack_seq=3266449517
      tcp_hdr_len=5 flags=ACK PSH
      res1=0H res2=0H window=31856 cksum=162H urgent=0
---> beginning of packet data dump <---
GET http://204.71.200.244/a/ya/yahoomail/promo1.gif HTTP/1.0
Referer: http://www.yahoo.com/
Connection: Keep-Alive
User-Agent: Mozilla/4.05 [en] (X11; U; Linux 2.1.103 i686)
Host: us.yimg.com
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg image/png
Accept-Language: en
Accept-Charset: iso-8859-1, *,utf 8

```

A POSITA would have been motivated to use this “keep-alive” functionality with each of the proxy caches (e.g., 110-1, 110-2, 115, and/or 117) and client 101-1 to help maintain TCP connections by preventing periods of inactivity and avoid unnecessarily expending resources to reestablish connections for each HTTP request. EX1002, ¶191. Thus, proxy redirector 104 (the claimed “first server”) sends

keep-alive messages to the proxy caches (the claimed “client devices”) and waits for response from each. EX1002, ¶191.

Limitation 10:

As explained in Section IX.C.1 and Limitation 1[c], proxy redirector 104 uses a stored criterion during the selection, such as “a metric such as least-loaded or round-robin [or] IP header information such as the origin server IP address.” EX1002, ¶193. A POSITA would have understood these criteria would be stored in proxy redirector 104 for the reasons discussed in Limitation [1p] (origin server IP address is stored in proxy redirector 104).

Limitation 12:

As explained in Section VII and IX.C.1, a POSITA would have understood that, “a metric such as least-loaded” is based on a response time when communicating. EX1004, 7:36-48; EX1002, ¶¶194-195.

Limitation 15:

As discussed for Limitations 1[p] and 1[b], proxy redirector 104 (the claimed “first server”) is a server that communicates over the internet with client 101-1 (the claimed “first client device”). EX1002, ¶196.

As explained in Section IX.C.1, a POSITA would have understood that proxy redirector 104 is a TCP/IP server that communicates over the Internet using TCP/IP, with client 101-1, and proxies 110-1, 110-2, 115, and/or 117 as TCP/IP client

devices. EX1002, ¶196. Cohen refers throughout to the use of TCP/IP, including with respect to proxy redirector 104 (“first server”) and the client devices. EX1002, ¶196. For example, “once the IP address of the origin server corresponding to the logical URL name is determined through the DNS look-up, proxy redirector 104, rather than establishing a TCP connection to the origin server at the determined IP address, transparently establishes a TCP connection between the client and a proxy.” EX1004, 7:12-17, Table 1 (showing proxy redirector 104 establishing a TCP connection with client 101-1 and proxy 110-1 through use of the “TCP header”). EX1002, ¶¶196-197. Thus, Cohen discloses or renders obvious proxy redirector 104 communicating with client 101-1 over the Internet using TCP/IP. EX1002, ¶204.

Limitation 16:

See Limitation 15. EX1002, ¶198 (communicating according to TCP standards).

Limitation 17:

See Limitation [1p]. EX1002, ¶199 (first content includes a web page; first content identifier includes a URL).

Limitation 18:

As explained in Limitation 2, proxy redirector 104 utilizes multiple proxies, such as proxy caches 110-1, 110-2, 115, and/or 117, each of which meet the construction of “second client device” and are addressable over the Internet using

the IP address that are selected from the list of IP addresses stored in proxy redirector 104. EX1002, ¶200.

Cohen discloses that the “invention relates to... a network for transparently intercepting client *web requests* and redirecting them to proxy caches.” EX1004, 1:6-10. Accordingly, it would have been obvious to a POSITA based on the disclosures and teachings of Cohen, as discussed Section IX.C.1 and in Limitation 1[a], for client 101-1 (the claimed “first client”) to send a second GET request that includes multiple URLs. EX1002, ¶201.

As explained for Limitation 1[c], the process of proxy redirector 104 selecting proxy caches 110-1, 110-2, 115, and/or 117 includes selecting the IP address of the proxy caches from the list of those IP addresses to route the URL through the proxy chain. EX1002, ¶202. Upon receiving the multiple URLs, proxy redirector 104 (the claimed “first server”) selects an IP address corresponding to a proxy cache from the group of IP addresses associated with two or more proxies (i.e., proxies 110-1, 110-2, 115, and/or 117) to forward the URLs to the selected proxies. EX1004, 7:43-48; 11:12-15; EX1002, ¶203. Proxy redirector may select a different IP address (the claimed “IP addresses of the multiple client devices”) corresponding to different proxy caches for each URL (e.g., 110-1 and 110-2). EX1002, ¶204.

As explained in Section IX.C.1 and for Limitation 1[d], proxy redirector 104 sends the URL to proxy cache 110-1 in response to the selecting. Communications

by proxy redirector 104 to each proxy cache use the stored IP address of the respective proxy cache as the “destination address” in accordance with IP protocols. EX1002, ¶204 (citing EX1015, 17/51; EX1016, 57/91). When proxy redirector 104 sends IP packets to proxy cache 110-1 and proxy cache 110-2, it uses the IP address of proxy cache 110-1 and proxy cache 110-2, respectively, (the claimed “selected IP address”) as the “destination address.” The “destination address” is part of the IP header and is thus sent with the IP packet. EX1002, ¶205 (citing EX1015, 17/51; EX1016, 57/91).

Limitation 19:

Cohen teaches an alternative where the proxy is modified so that the proxy cache knows the client IP address, (the claimed first client). EX1002, ¶206; EX1004 16:44-51. In this way the packets from the proxy cache are sent directly back to the client. EX1002, ¶206; *see* EX1004 16:67–17:1. A POSITA would understand that one obvious way to do this would be for the packets to sent directly from the proxy cache back to the client on another TCP connection, using the IP address of the proxy cache as the source address in the IP packet. A POSITA would be motivated to do this to reduce delay. EX1004 17:1-3. In this manner, the selected IP address is sent to the first client. EX1002, ¶206.

Limitation 21:

Client 101-1 (the claimed “first client device”) is identified by a first IP address—specifically its own IP address—in sessions with proxy redirector 104. EX1002, ¶207; *see also* EX1004, Table 1.

TABLE 1

Arriving packet:

```

---> beginning of packet header dump <---
---> IP header: version=4 hdr_len=5 TOS=0 pkt_len=346 id=60276
      frag_off=4000H TTL=64 protocol=6 cksum=1792H
      saddr=135.104.25.243 daddr=204.71.200.244
---> TCP header: sport=1273 dport=80 seq=2189084427 ack_seq=3266449517
      tcp_hdr_len=5 flags=ACK PSH
      res1=0H res2=0H window=31856 cksum=162H urgent=0
---> beginning of packet data dump <---
GET /a/ya/yahoomail/promo1.gif HTTP/1.0
Referer: http://www.yahoo.com/
Connection: Keep-Alive
User-Agent: Mozilla/4.05 [en] (X11; U; Linux 2.1.103 i686)
Host: us.yimg.com
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg image/png
Accept-Language: en
Accept-Charset: iso-8859-1, *,utf 8
Modified to:

```

```

---> beginning of packet header dump <---
---> IP header: version=4 hdr_len=5 TOS=0 pkt_len=367 id=60276
      frag_off=4000H TTL=64 protocol=6 cksum=1792H
      saddr=135.104.25.245 daddr=135.104.25.31
---> TCP header: sport=5000 dport=7000 seq=2189084427 ack_seq=3266449517
      tcp_hdr_len=5 flags=ACK PSH
      res1=0H res2=0H window=31856 cksum=162H urgent=0
---> beginning of packet data dump <---
GET http://204.71.200.244/a/ya/yahoomail/promo1.gif HTTP/1.0
Referer: http://www.yahoo.com/
Connection: Keep-Alive
User-Agent: Mozilla/4.05 [en] (X11; U; Linux 2.1.103 i686)
Host: us.yimg.com
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg image/png
Accept-Language: en
Accept-Charset: iso-8859-1, *,utf 8

```

Limitation 22:

Cohen discloses or renders obvious this limitation. EX1002, ¶208. As explained in Sections VII and IX.C.1, a POSITA would understand Cohen's disclosure of web requests and responses to disclose HTTP requests and responses. EX1002, ¶208. For example, Cohen explains that "HTTP connection requests are generally directed to port 80 of an origin server." EX1004, 3:3-4. Table 1 of Cohen and the surrounding text illustrates an example "HTTP" request directed to a web server at www.yahoo.com with a particular IP address. EX1004, Table 1, 10:33-11:23; EX1002, ¶209.

Limitation 23:

As discussed for Limitations 1[p]-1[f], Cohen discloses proxy redirector 104 performing the steps of Claim 1 in a sequential manner. EX1002, ¶210.

Limitation 24:

Cohen discloses or renders obvious this limitation. *See* Limitation [1a]; EX1002, ¶211. Cohen discloses that the proxy redirector ("first server") may be a Level 4 switch that also may be combined into the same unit as a router. EX1004, 6:23-31. The computer functions may be carried out by "software in any form, including, therefore, firmware, microcode or the like, combined with appropriate circuitry for executing that software to perform the function." *Id.*, 18:1-3. The computer functions are executed by processors. *Id.*, 17:36-56. Thus, the processors

disclosed in Cohen would execute computer instructions and perform the recited functions.

Further, a POSITA would understand that, as is the nature of computing devices, proxy redirector 104 would include a processor to execute the computer instructions from the software and perform the steps of claim 1. EX1002, ¶212.

Limitation 25:

Cohen discloses or renders obvious this limitation. *See* Limitations [1a], [24]; EX1002, ¶213. The processes disclosed in Cohen “may be substantially represented in computer readable medium and so executed by a computer or processor:” EX1004, 17:36-41.

Further, a POSITA would understand that, as is the nature of computing devices, proxy redirector 104 would include a processor and a non-transitory computer readable medium containing computer instructions that are to be executed by the processor to cause the processor to perform the steps. EX1002, ¶214.

D. Ground 4 – Cohen in View of RFC 1122 Renders Obvious Claims 6-9,

With Ground 3 as predicate, Challenged Claims 6-9 would have been obvious to a POSITA based upon Cohen in view of RFC 1122 (*see* §IX.B.1).

1. Motivation to combine

A POSITA would have been motivated to use the TCP keep-alive functionality of RFC 1122 as discussed in Section IX.B.1, above, when maintaining

the TCP connections as disclosed by Cohen. Both references cover connecting Internet hosts over the Internet. EX1002, ¶216. RFC 1122 describes an Internet host as “A host computer, or simply “host,” is the ultimate consumer of communication services.” EX1006, Section 1.1.1, page 6. Cohen describes client requesting objects stored in servers connected to the Internet. EX1002, ¶216; EX1004 6:33-39. Cohen specifically teaches that the proxy redirector should only select for available proxy caches. EX1002, ¶216; EX1004 7:42-46.

A POSITA would have recognized the benefits of Cohen’s proxy redirector exchanging TCP keep-alives with the proxy caches 110-1, 110-2, 117 to know if these proxy caches were available. EX1002, ¶217. One benefit, TCP keep-alives help detect when “a client crashes or aborts a connection during a network failure” and thus prevent server applications from “hang[ing] indefinitely and consum[ing] resources unnecessarily.” EX1006, 102/116. For example, following the establishment of connections between the proxy redirector and the proxy caches, one of the proxy caches could crash. EX1002, ¶217. This could lead to the situation where the proxy redirector sends Internet traffic to a proxy cache that no longer operational. *Id.* This could lead to lost messages and unnecessarily expend resources keeping the “dead” TCP connection open. *Id.* Another benefit a POSITA would have known of is that TCP keep-alives could also prevent periods of inactivity

between the proxy redirector and the proxy caches from causing the TCP connection between them to terminate. *Id.* Which again could result in lost messages.

RFC 1122 does specify some potential disadvantages of TCP keep-alives, but a POSITA would have recognized that these disadvantages are outweighed by the benefits listed above in a system such as Cohen's, which explicitly disclose the need of keep-alive. EX1002, ¶218.

2. Modification of Cohen in view of RFC 1122

A POSITA would have modified Cohen so that the TCP connections between the proxy redirector and the proxy caches would periodically communicate by exchanging TCP keep-alive messages with each other over established TCP connections. EX1002, ¶219. A "keep-alive" mechanism was a well known technique which periodically probes the other end of a connection when the connection is otherwise idle, even when there is no data to be sent. EX1006, 102/116. In the proxy chain comprising a client computer 101 -> proxy redirector 104 -> proxy cache 117, the proxy redirector 104 would periodically send client computer 101 and proxy caches 117 keep-alive messages over the TCP connections between them. EX1002, ¶219. Further, in accordance with RFC 1122, the proxy 17 would wait for an "acknowledgment" from proxy caches, confirming that the connection is still live. *See* EX1006, 102/116

Such as modification would be simple for a POSITA, as all that is required is to send a probe segment designed to elicit a response from the peer TCP periodically. EX1006, 102/116.

3. Element-by-Element Analysis

Limitation 6:

Cohen-RFC 1122 renders obvious this limitation. EX1002, ¶221. As described above Sections IX.D.1-2, proxy redirector 104 would periodically send client computer 101 and proxy caches 117 keep-alive messages over the TCP connections and would wait for an “acknowledgment” from proxy caches, confirming that the connection is still live.

Limitation 7:

As described above in Ground 3, Limitation 2, Cohen discloses how the proxy caches are identified over the Internet using a respective IP address. The addition of TCP keep-alives from RFC1122 would allow a POSITA to identify if a TCP connection were alive, and if it was alive, A POSITA would know the proxy cache was available and online. *See* EX1002, ¶222; EX1006, 102/116.

As described in Ground 3, Limitation 7, when there exists unacknowledged probes less than the value, `tcp_keepalive_probes`, the connection is neither alive or dead. EX1002, ¶222. In this state it would be prudent for the POSITA to remove

this proxy cache from the list of available proxies, as opposed to leaving it on the available proxy cache list in case the proxy cache itself was dead. *Id.*

If a keep-alive probe is subsequently received by proxy redirector, the first server, a POSITA would then place the proxy cache from which the tcp-keepalive probe was received from back on the list of available proxies. EX1002, ¶223.

Limitation 8:

As discussed for Limitation 7, it would have been obvious to remove the IP address of a non-response. If the proxy redirector determines a connection is not active, then it may consider the connection terminated and remove the IP address of the terminated proxy caches from the list of IP addresses of proxy caches devices under active connections. EX1002, ¶224. Thus, proxy redirector maintains a list of IP addresses whose respective proxy caches are current returning TCP keep-alives. *Id.*

Limitation 9:

RFC 1122 teaches that the TCP keep alive mechanism **periodically** probes the connection. EX1006, 102/116; EX1002, ¶225.

X. CONCLUSION

For the above reasons, all Challenged Claims are invalid.

Dated: December 5, 2023

/Eagle H. Robinson/

Eagle H. Robinson
Reg. No. 61,361

Lead Counsel for Petitioner

CERTIFICATE OF WORD COUNT

Pursuant to 37 C.F.R. § 42.24(a)(i), the undersigned certifies that this Paper—exclusive of the table of contents, mandatory notices under § 42.8, certificate of service, and this certificate of word count—includes 13,950 words.

/Eagle H. Robinson/
Eagle H. Robinson (Reg. No. 61,361)

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. § 42.6(e) and 37 C.F.R. § 42.105(a), the undersigned certifies that on December 5, 2023, a complete copy of this Petition for *Inter Partes* Review, Petitioner's power of attorney, and EX1001-EX1023 were served on Patent Owner at the correspondence addresses of record listed below by FedEx®:

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

/Eagle H. Robinson/
Eagle H. Robinson (Reg. No. 61,361)