

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CISCO SYSTEMS, INC.,
Petitioner

IPR2025-00188
U.S. Patent No. 8,982,691

**PETITION FOR *INTER PARTES* REVIEW
UNDER 35 U.S.C. § 312 AND 37 C.F.R. § 42.104**

TABLE OF CONTENTS

PETITIONER’S EXHIBIT LIST	5
I. INTRODUCTION	7
II. GROUNDS FOR STANDING.....	8
III. NOTE.....	8
IV. SUMMARY OF THE ’691 PATENT	8
A. Overview of the ’691 patent.....	8
B. Prosecution History of the ’691 patent.....	10
V. LEVEL OF ORDINARY SKILL IN THE ART.....	12
VI. CLAIM CONSTRUCTION	12
VII. RELIEF REQUESTED AND REASONS FOR THE REQUESTED RELIEF	13
VIII. DISCRETIONARY DENIAL WOULD BE INAPPROPRIATE.....	13
A. Discretionary denial under <i>Fintiv</i> is not appropriate.	13
1. No evidence regarding a stay	13
2. Parallel proceeding trial date	14
3. Investment in the parallel proceeding.....	14
4. Overlapping issues with the parallel proceeding	15
5. Identity of parties	15
6. Other circumstances.....	15
B. Discretionary denial under 35 U.S.C. § 325(d) is not appropriate. ...	16

C.	Discretionary denial under <i>General Plastic</i> is not appropriate.....	17
IX.	IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE....	17
A.	Challenged Claims and Statutory Grounds for Challenges	17
B.	Statement of Material Facts.....	18
C.	Ground 1: Claims 1-10 are Obvious over EDC_525, EDC_892, and Hanif.	21
1.	EDC_525.....	21
2.	EDC_892.....	25
3.	Hanif.....	26
4.	Reasons to Combine EDC_892 with EDC_525.	27
5.	Reasons to Combine Hanif with EDC_525	29
6.	Claim 1	33
7.	Claim 2.....	70
8.	Claim 3	71
9.	Claim 4.....	72
10.	Claim 5	73
11.	Claim 6.....	74
12.	Claim 7	77
13.	Claim 8.....	77
14.	Claim 9	77
15.	Claim 10.....	77

D.	Ground 2: Claims 1-10 are Obvious over EDC_525, EDC_892, Hanif, and Li.....	77
1.	Li	77
2.	Reasons to Combine Li with EDC_525.....	78
3.	Claim 1	83
4.	Claims 2-5	86
5.	Claim 6.....	86
6.	Claims 7-10.....	87
X.	CONCLUSION.....	87
XI.	MANDATORY NOTICES	88
A.	Real Party-in-Interest	88
B.	Related Matters.....	88
C.	Lead and Back-up Counsel and Service Information	88
XII.	CLAIMS APPENDIX	90
	CERTIFICATE OF WORD COUNT	94
	CERTIFICATE OF SERVICE	95

PETITIONER’S EXHIBIT LIST

Ex.1001	U.S. Patent No. 8,982,691
Ex.1002	Prosecution History of U.S. Patent No. 8,982,691.
Ex.1003	Declaration of Dr. Henry Houh under 37 C.F.R. § 1.68.
Ex.1004	<i>Curriculum Vitae</i> of Dr. Henry Houh.
Ex.1005	U.S. Patent Pub. No. 2004/0218525, (“EDC_525”).
Ex.1006	U.S. Patent Pub. No. 2004/0246892, (“EDC_892”).
Ex.1007	Provisional Application 60,328,087.
Ex.1008	U.S. Patent No. 8,913,481.
Ex.1009	“Dynamic Path Management with Resilience Constraints under Multiple Link Failures in MPLS/GMPLS Networks,” Park et al, IEEE 2008.
Ex.1010	“Optimal and Guaranteed Alternative LSP for Multiple Failures,” Hundessa et al., IEEE 2004.
Ex.1011	U.S. Patent No. 7,835,267.
Ex.1012	U.S. Patent No. 9,559,947.
Ex.1013	U.S. Patent Pub. No. 2009/0292943, (“Hanif”).
Ex.1014	Microsoft Computer Dictionary, (1999).
Ex.1015	U.S. Patent No. 7,821,951.
Ex.1016	C++ Inside and Out, Eckel (1993).
Ex.1017	Programming Microcontrollers in C, (Sickle) 1994.
Ex.1018	RFC3945.
Ex.1019	RFC4090.
Ex.1020	U.S. Patent No. 7,672,226.
Ex.1021	U.S. Patent No. 7,626,925.
Ex.1022	Traffic Engineering with MPLS, Osborne (2003).
Ex.1023	Fault-Tolerant IP and MPLS Networks, Hussain (2004).

Ex.1024	U.S. Patent Pub. No. 2009/0219806A1.
Ex.1025	U.S. Patent Pub. No. 2006/0159009A1.
Ex.1026	MPLS Traffic Engineering Path Link and Node Protection Configuration Guide, Cisco Press, 2011.
Ex.1027	Protection Performance Components in MPLS Networks, Calle (2004).
Ex.1028	U.S. Patent No. 7,616,637.
Ex.1029	RFC3031.
Ex.1030	Certified English Translation of CN101645848A.
Ex.1031	CN101645848A.
Ex.1032	Websters New World Dictionary of Computer Terms (2000)
Ex.1033	Computer Desktop Encyclopedia (2001)
Ex.1034	The C language interface to the SQLite library
Ex.1035	University of Hawaii, EE160 Book, Chapter 9, Two Dimensional Arrays
Ex.1036	Functional C (1999)
Ex.1037	UT Austin Lecture_2010
Ex.1038	A Complete Guide to C++
Ex.1039	Introduction to Pro_C
Ex.1040	Complaint - <i>WSOU Investments, LLC v. Cisco Systems, Inc.</i> EDTX-2-24-cv-00332.
Ex.1041	Federal Court Trial Statistics – June 30, 2024.
Ex.1042	Docket Control Order - <i>WSOU Investments, LLC v. Cisco Systems, Inc.</i> , No. 2-24-cv-00332, (E.D. Tex. August 5, 2024).

I. INTRODUCTION

Pursuant to 35 U.S.C. §§ 311, 314(a), and 37 C.F.R. § 42.100, Cisco Systems, Inc. (“Petitioner”) respectfully requests that the Board review and cancel as unpatentable under (pre-AIA) 35 U.S.C. §103(a) claims 1-10 (the “Challenged Claims”) of U.S. Patent No. 8,982,691 (“’691 patent,” Ex.1001).

The ’691 patent generally relates to multiprotocol label switching-transport (“MPLS”) networks. Ex.1001, 1:5-10. The alleged novelty, according to the ’691 patent, is the use of multiple backup paths to protect against multiple network failures. Ex.1001, 1:42-46, 2:1-27; Ex.1002, 16-22, 41.

This idea was not novel. When the ’691 patent application was filed, the use of multiple backup paths to protect from multiple failures was already known in the art. Ex.1003, ¶¶45-46, 60. For example, U.S. Patent Publication No. 2004/0218525 to Elie-Dit-Cosaque (“EDC_525,” Ex.1005) discloses that “having multiple backup paths advantageously decreases the probability of simultaneous interruption” and protects against “multiple failures in a network.” Ex.1005, [0034], [0008], Abstract.

Accordingly, Petitioner respectfully requests that the Board institute *inter partes* review (“IPR”) and cancel as unpatentable the Challenged Claims of the ’691 patent.

II. GROUNDS FOR STANDING

Petitioner certifies that the '691 patent is eligible for IPR and that Petitioner is not barred or estopped from requesting IPR challenging the patent claims. 37 C.F.R. § 42.104(a).

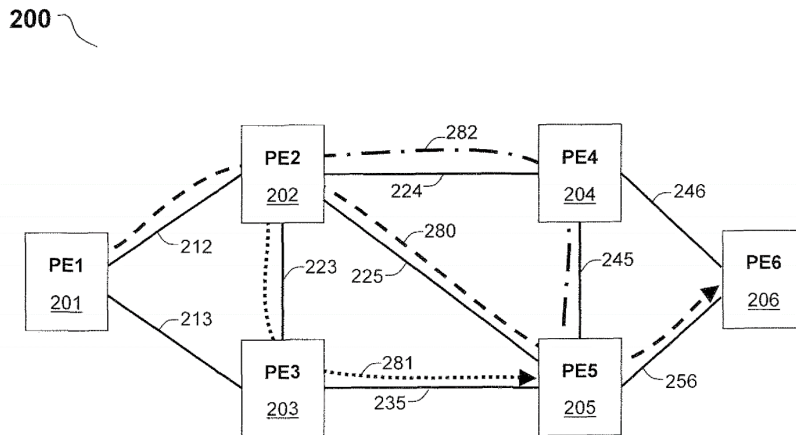
III. NOTE

Petitioner cites to exhibits' original page numbers whenever feasible. Emphasis in quoted material has been added.

IV. SUMMARY OF THE '691 PATENT

A. Overview of the '691 patent

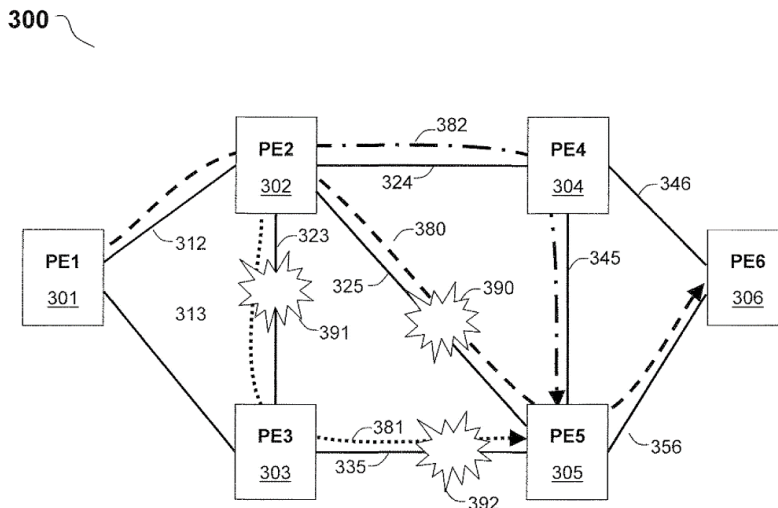
The '691 patent pertains to MPLS networks. Ex.1001, 1:7-10, Abstract. The purported novelty of the '691 patent is that it utilizes multiple backup label switch paths ("LSPs") "to accommodate double-fault scenarios." Ex.1001, 1:63-2:27; Ex.1003, ¶51. The '691 patent's FIG. 2 illustrates an example where an MPLS network has three LSP paths: namely, primary LSP 280 (working path), a Bypass LSP 281 (first backup path), and a Backup LSP 282 (second backup path) that provides protection in case the other paths fail. Ex.1001, 1:42-46, 5:38-46; Ex.1003, ¶52.



Ex.1001, FIG. 2.

Fig. 2

FIG. 3 illustrates a scenario where LSP 380 (primary path) has a fault (390) and the Bypass LSP 381 (first backup path) simultaneously also has a fault (391, 392). The Backup LSP 382 (second backup path) provides protection. Ex.1001, 5:47-58; Ex.1003, ¶¶53-54.



Ex.1001, FIG. 3.

Fig. 3

The '691 patent's FIG. 4 illustrates a method for providing paths for

multiple fault protection. Ex.1001, 6:4-50; Ex.1003, ¶¶55-58.

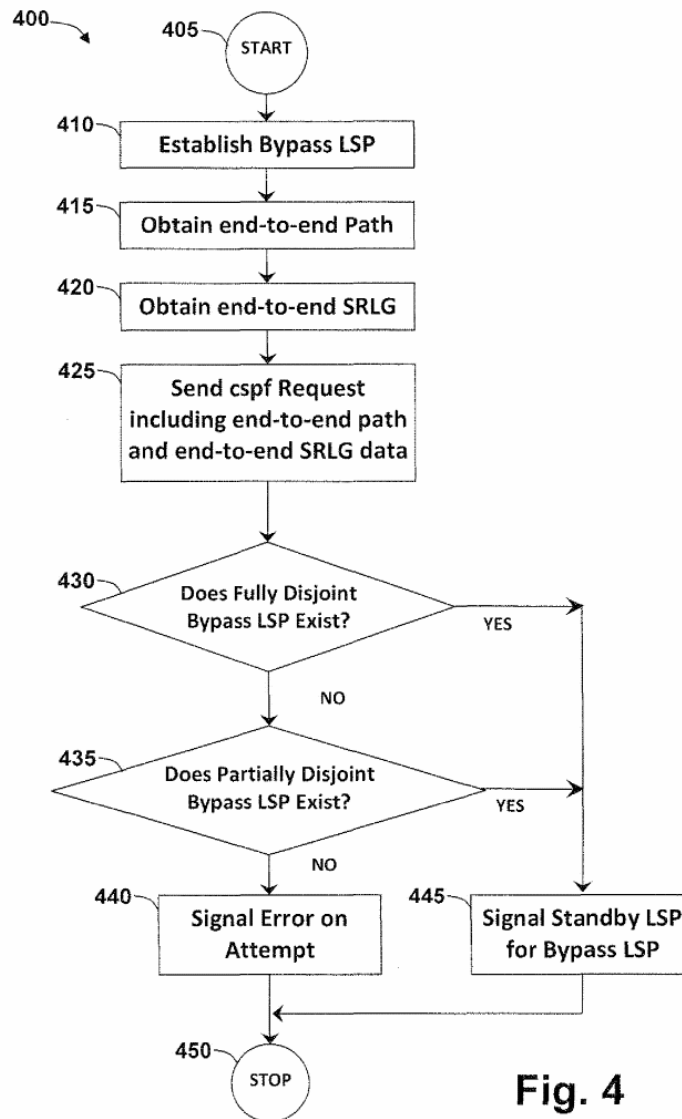


Fig. 4

Ex.1001, FIG. 4.

B. Prosecution History of the '691 patent.

During prosecution of the '691 patent, the Applicants and the Examiner agreed to the following claim amendments, after which the claims issued:

1. (Currently Amended) A method performed by a network processor of a Multiprotocol Label Switching (MPLS) label switch router for providing a Backup Label Switched Path (LSP) to ~~an already established~~ a Bypass LSP already established for a Protected Primary LSP, the method comprising the steps of:

protecting the Primary LSP against dual failures, comprising:

establishing ~~[[a]]~~ the Bypass LSP for the Protected Primary LSP having a Point of Local Repair node and a Merge Point node;

obtaining the nodes traversed by an end-to-end path of said Bypass LSP from said Point of Local Repair Node to said Merge Point node;

generating a request to a path calculator using the nodes traversed by said end-to-end path of said Bypass LSP for a disjoint path connecting said Point of Local Repair Node to said Merge Point node;

receiving a response from said path calculator; and

~~in the event that~~ in response to determining that a fully disjoint path connecting said Point of Local Repair Node to said Merge Point node is available, ~~then~~ signaling, to at least one other MPLS label switch router, said fully disjoint path as ~~[[a]]~~ the Backup LSP to said Bypass LSP.

Ex.1002, 16-17. The Examiner appears to have allowed the claims based on Applicant's argument that the amendments require that the Protected LSP, Bypass LSP, and Backup LSP are all in existence at the same time, "and not calculated only upon a failure":

The claims have been amended to clarify that the Bypass LSP is “already established for a Protected LSP,” as described in the Specification at, for example, paragraphs [0001]-[0002], [0004]-[0005], [0034], and [0042], to further describe the subject matter of the invention, and emphasize that the Protected LSP and Bypass LSP are “established” at the same time as the Backup LSP. Thus it may be understood that the Bypass LSP is already established for a Protected LSP, and not calculated only upon failure of a first connection as in Canali.

Ex.1002, 41.

As shown below, the added features were known and it would have been obvious to combine the prior art as claimed. Ex.1003, ¶¶59-60.

V. LEVEL OF ORDINARY SKILL IN THE ART

A Person of Ordinary Skill in The Art (“POSITA”) on September 28, 2012, would have been someone knowledgeable and familiar with network communications and MPLS techniques available at the time. Ex.1003, ¶¶25-26. That person would have a bachelor’s degree in computer science, computer engineering, electrical engineering, or equivalent training, and approximately two years of experience working in the field of network communications and would be knowledgeable regarding MPLS techniques. Lack of work experience can be remedied by additional education, and vice versa. Ex.1003, ¶¶26-27.

VI. CLAIM CONSTRUCTION

Claim terms in IPR are construed according to their “ordinary and customary meaning” to those of skill in the art. 37 C.F.R. § 42.100(b). *Phillips v. AWH Corp.*,

415 F.3d 1303, 1313 (Fed. Cir. 2005) (en banc). Petitioner submits that, for the purposes of this proceeding and the grounds presented herein, no claim term requires express construction. *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017). Ex.1003, ¶¶61.

VII. RELIEF REQUESTED AND REASONS FOR THE REQUESTED RELIEF

Petitioner asks that the Board institute a trial for IPR and cancel the Challenged Claims in view of the analysis below. Ex.1003, ¶¶62-66.

VIII. DISCRETIONARY DENIAL WOULD BE INAPPROPRIATE

A. Discretionary denial under *Fintiv* is not appropriate.

The six factors considered for § 314 discretionary denial strongly favor institution given Petitioner's timely filing and the compelling merits of this case. *See Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 (PTAB Mar. 20, 2020) (precedential).

1. No evidence regarding a stay

No motion to stay has been filed, so the Board should not infer the outcome of such a motion. *Sand Revolution II LLC v. Continental Intermodal Group – Trucking LLC*, IPR2019-01393, Paper 24 at 7 (PTAB June 16, 2020) (informative); *see also Dish Network L.L.C. v. Broadband iTV, Inc.*, IPR2020-01359, Paper 15 (Feb. 12, 2021). Thus, this factor is neutral.

2. Parallel proceeding trial date

The co-pending district court case was filed in the Eastern District of Texas on May 6, 2024. Ex.1032. The most recent statistics currently available for the Eastern District of Texas show a median time-to-trial of 21.6 months. Ex.1041, 35. Thus, the estimated trial date is late March 2026.¹ The Board’s final written decision is expected soon after, in April or May 2026. Given the proximity of the Board’s expected Final Written Decision and the estimated trial date, this factor is at least neutral. *See e.g., Cisco Systems Inc., v. Video Solutions Pte. Ltd.*, IPR2024-00194, Paper 7 at 10 (PTAB March 28, 2024) (granting institution and finding factor two neutral where “trial is likely to begin within approximately one month of [the Board’s] final written decision”).

3. Investment in the parallel proceeding

The co-pending litigation is in its early stages, with major activities including a *Markman* hearing and the close of discovery scheduled after the Board’s expected decision on institution. *See* Ex.1042, 3-4 (Claim Construction

¹ The docket control order sets the case for trial on March 2, 2026. Ex.1042, 1. However, “a court’s general ability to set a fast-paced schedule is not particularly relevant...where, like here, the forum itself has not historically resolved cases so quickly.” *In re Apple Inc.*, 979 F.3d 1332, 1344 (Fed. Cir. 2020).

hearing on September 3, 2025; close of fact discovery on October 14, 2025; close of expert discovery on November 17, 2025). Moreover, Petitioner has acted with diligence to prepare this Petition approximately six months after Petitioner was served with a complaint. Under *Fintiv*, Petitioner’s prompt filing “weigh[s] against exercising the authority to deny institution.” *Fintiv*, Paper 11 at 11.

4. Overlapping issues with the parallel proceeding

Claims 1-5, challenged here, are not asserted in the district court litigation. Petitioner also offers the following stipulation: Petitioner stipulates that if trial is instituted, it will not pursue the same grounds in the district court litigation. *See Sand Revolution II, LLC v. Cont’l Intermodal Group-Trucking LLC*, IPR2019-01393, Paper 24 at 12 (PTAB June 16, 2020). Consequently, this factor favors institution.

5. Identity of parties

Petitioner is a defendant in the litigation. That is true of most petitioners in IPR proceedings. Accordingly, this factor should not be a basis for denying institution.

6. Other circumstances

The prior art presented in this Petition presents a compelling case on the merits. During prosecution, the claims of the ’691 patent were amended and the Applicants argued that prior art does not teach that a Protected LSP, a Bypass LSP,

and a Backup LSP are all in existence at the same time, as opposed to calculating the Backup LSP upon failure. Ex.1002, 41. As explained in detail below, these ideas (and the remaining claim limitations) were disclosed in other prior art not reviewed by the Examiner.

The Board's current policy is to "not deny institution of an IPR or PGR under *Fintiv* (i) when a petition presents compelling evidence of unpatentability." Director's Memorandum dated June 21, 2022 ("Memo"), 9. "Compelling, meritorious challenges are those in which the evidence, if unrebutted in trial, would plainly lead to a conclusion that one or more claims are unpatentable by a preponderance of the evidence." Memo, 4. Here, the Petition plainly shows that the concepts added to gain allowance were known in the art.

Thus, the evidence of unpatentability is compelling here, and the PTAB should not deny institution under *Fintiv*.

B. Discretionary denial under 35 U.S.C. § 325(d) is not appropriate.

Denial under § 325(d) is not warranted because the challenges presented in this petition are neither cumulative nor redundant to the prosecution of the '691 patent. The Examiner did not consider any of the references relied upon in the grounds of unpatentability presented below. Moreover, the challenges in this petition are non-cumulative because the prior art presented here describes the claim limitations that were amended to gain allowance; namely, the prior art discloses

that a Protected LSP, a Bypass LSP, and a Backup LSP are all in existence at the same time, as opposed to calculating the Backup LSP upon failure. *See* Ex.1002, 16-22, 41. Accordingly, the Board should not exercise its discretion to deny this petition under § 325(d).

C. Discretionary denial under *General Plastic* is not appropriate.

The '691 patent has not been challenged in any prior IPR petition, so none of *General Plastic* discretionary institution factors apply to this Petition. *See General Plastic Indus. Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19 at 16 (PTAB Sept. 6, 2016) (Section II.B.4.i. precedential).

IX. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE

A. Challenged Claims and Statutory Grounds for Challenges²

Grounds	Claims	Basis: 35 U.S.C. § 103 (Pre-AIA) over
#1	1-10	EDC_525, EDC_892, and Hanif
#2	1-10	EDC_525, EDC_892, Hanif, and Li

- U.S. Patent Publication No. 2004/0218525 to Elie-Dit-Cosaque (“EDC_525,” Ex.1005) was filed on 4/29/2003, published on 11/4/2004.

² Petitioner relies on the teachings, and not on a physical incorporation of elements. *See In re Mouttet*, 686 F.3d 1322, 1332 (Fed. Cir. 2012); *In re Etter*, 756 F.2d 852, 859 (Fed. Cir. 1985); Ex.1003, ¶64.

- U.S. Patent Publication No. 2004/0246892 to Elie-Dit-Cosaque (“EDC_892,” Ex.1006) was filed on 11/29/2001, published on 12/9/2004.
- U.S. Patent Publication No. 2009/0292943 to Hanif et al. (“Hanif,” Ex.1013) was filed on 8/2/2007, published on 11/26/2009.
- Chinese Patent Publication No. CN101645848A to Li et al. (Certified English Translation Ex.1030, Original Ex.1031, “Li”) was published on 2/10/2010.

Each reference is prior art under (pre-AIA) 35 U.S.C. § 102(b).³

Petitioner’s analysis cites additional prior art to support assertions regarding what a POSITA would have known or understood at the time. *See Yeda Research v. Mylan Pharm. Inc.*, 906 F.3d 1031, 1041-1042 (Fed. Cir. 2018); 37 C.F.R. § 42.104(b); *see also K/S HIMPP v. Hear-Wear Techs., LLC*, 751 F.3d 1362, 1365-66 (Fed. Cir. 2014); *Arendi S.A.R.L. v. Apple Inc.*, 832 F.3d 1355, (Fed. Cir. 2016).

B. Statement of Material Facts

- 1) As of the 2012 earliest claimed priority date, MPLS networks were well known and standardized. Ex.1003, ¶¶33-34; Ex.1028, 3:30-33; Ex.1029, *generally*.

³ The references also qualify as prior art under (post-AIA) 35 U.S.C. § 102(a)(1).

- 2) Although MPLS was originally developed for IP networks, it was extended to optical networks as GMPLS. Ex.1003, ¶¶34, 107; Ex.1028, 3:26-33. Ex.1011, 1:25-27; Ex.1018, 1.
- 3) Using one or more preestablished backup paths to provide protection was known in the art. Ex.1003, ¶¶35-36; Ex.1005, [0009]; Ex.1006, Abstract; Ex.1007, 1; Ex.1022, 294; Ex.1022, 293.
- 4) It was known to establish MPLS paths by setup signaling. Ex.1011, 3:7-9, 4:7-9; Ex.1018, 37.
- 5) Typically, multiple backup paths were used to protect from multiple failures. Ex.1003, ¶46; Ex.1005, [0034]; Ex.1010, Abstract; Ex.1012, 9:28-32.
- 6) Protection techniques were generally grouped into two categories: (1) “global” or “path” protection, which uses a preestablished path to protect a primary path end-to-end from source to the destination, and (2) “local” protection, which uses one or more preestablished backup paths to protect part of a primary path (e.g., using link protection or node protection). Ex.1003, ¶¶38-40; Ex.1022, 293-295; Ex.1023, 278; Ex.1022, 295.
- 7) Global (or path) protection was known to be less scalable because it requires a dedicated 1:1 relationship between the primary and the backup path, whereas local protection was more scalable because it allowed sharing backup paths. Ex.1003, ¶¶41-42, 145; Ex.1022, 295.

8) One common type of local protection is known as Fast ReRoute (“FRR”).

Ex.1003, ¶43; Ex.1023, 281; *see also* Ex.1001, 1:16-24.

9) FRR (or local protection) was recognized as providing numerous benefits, including (1) quick rerouting on to one or more preestablished backup LSPs in case of failure, (2) limiting packet loss, (3) reducing traffic disruption during failure, among others. Ex.1003, ¶¶44; Ex.1023, 278, 281; Ex.1022, 295; Ex.1011, 1:38-45.

10) In FFR (or local protection) the backup path begins at a node called Point of Local Repair (“PLR”) and terminates at a node called Merge Point (“MP”), where the backup path rejoins the protected path. Ex.1003, ¶¶45, 148; Ex.1022, 296; *see also* Ex.1001, 1:42-46.

11) Typically path computations were performed either at a central node or distributed over network nodes that included program instructions executable by a processor. Ex.1003, ¶¶49, 111-112; Ex.1005, [0025]; Ex.1012, 12:37-13:10, FIGS. 9-10; Ex.1008, 20:41-43; Ex.1012, 3:31-33.

12) The processor executing instructions typically generates a request, which was conventionally referred to in the art as a function call, that has input arguments or parameters needed by the function. Ex.1003, ¶¶49-50, 111-112, 170, 179; Ex.1012, 3:13-17; Ex.1014, 199-200; Ex.1016, 96; Ex.1017, 37; Ex.1033, 380.

- 13) Typically, after the called function performs its operation, results are returned. Ex.1003, ¶¶49-50, 111-112, 170, 179; Ex.1016, 98; Ex.1033, 380.
- 14) It was conventional, indeed standardized, for backup paths to have an associated priority. Ex.1003, ¶¶47-48; Ex.1025, [0043]-[0044]; Ex.1026, 44-45; Ex.1030, 6-7.
- 15) It was known to use a path's priority to identify the sequence that it would be used in case of failure. Ex.1003, ¶¶47-48; Ex.1026, 44-45; Ex.1030, 6-7.
- 16) The terms “backup path,” “protection path,” “bypass path,” “backup tunnel,” “protection tunnel,” and “bypass tunnel” are substantially synonymous. Ex.1003, ¶¶37, 119, n.1; Ex.1022, 293, 296; Ex.1023, 281; Ex.1005, [0034].

C. Ground 1: Claims 1-10 are Obvious over EDC_525, EDC_892, and Hanif.

1. EDC_525

EDC_525 generally discloses techniques for providing multiple backup paths to protect from multiple failures in a generalized multi-protocol label switched (GMPLS) optical transport network. Ex.1005, [0001]-[0009], [0022], Abstract, FIGS. 1, 4, 6-9, Claims 1, 31. A POSITA would have recognized that GMPLS is a version of MPLS designed for optical networks, among others. *See*

e.g., Ex.1003, ¶¶33-34, 68-69, 107; *see also* Material Facts #1-2; Ex.1011, 1:25-27; Ex.1012, 15:43-47; Ex.1018, 1; Ex.1028, 3:26-33.

EDC_525 discloses that a network administrator manager (“NAM”) and a quality monitor (“QM”) compute paths in an optical network 100. The NAM and QM “may be disposed centrally or distributed over one or more nodes.” Ex.1005, [0025]; *see also* Ex.1005, [0042]. EDC_525 further explains that its nodes perform routing and switching. Ex.1005, [0025]-[0026]; Ex.1003, ¶70.

EDC_525 illustrates at FIG. 2 an exemplary node that is responsible for control in the GMPLS network. Ex.1003, ¶71.

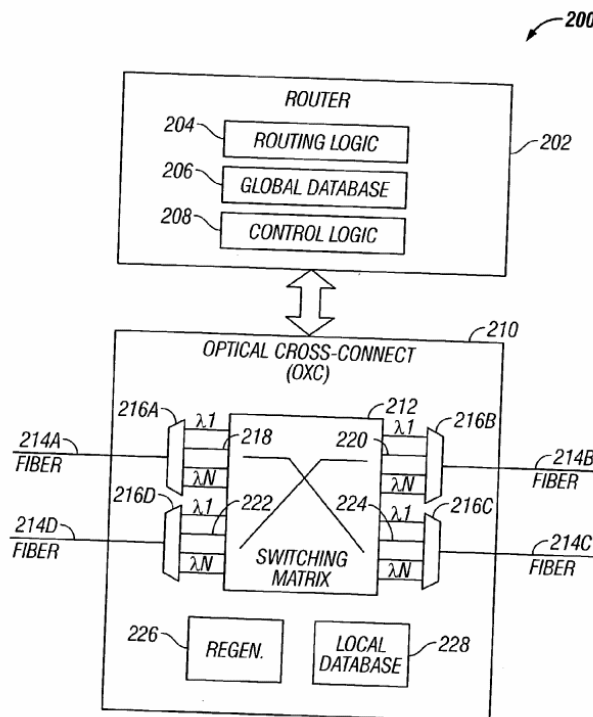


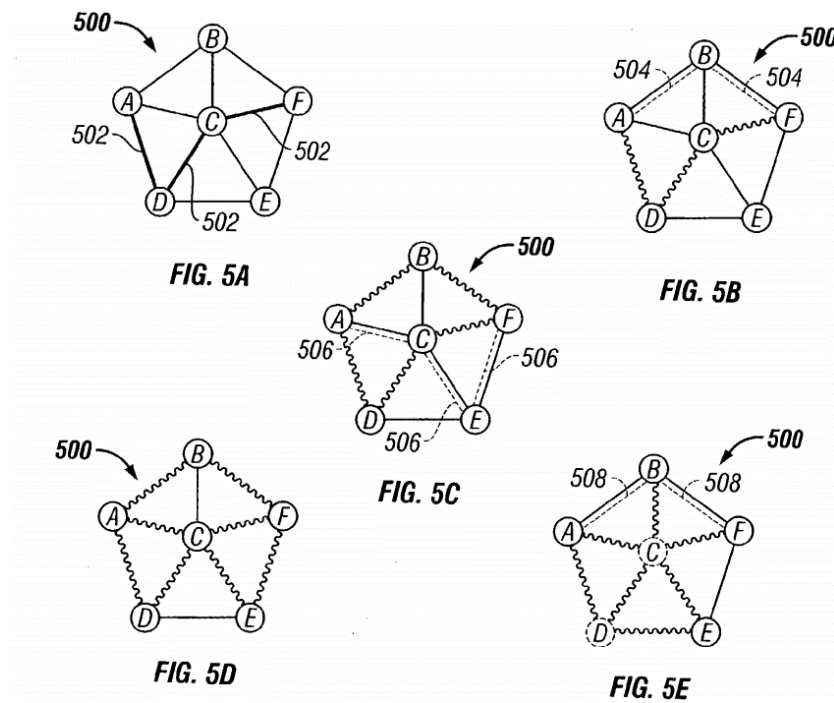
FIG. 2

Ex.1005, FIG. 2.

Like the '691 patent, EDC_525 “address[es] the issue of correlated multiple failures” by disclosing that its node (either centrally or distributed) performs path calculations and provides multiple backup paths for protection “under a scenario of multiple failures.” Ex.1005, [0007]-[0008], [0025]-[0026], [0045], Abstract; Ex.1003, ¶72.

EDC_525’s node has processor-accessible medium with instructions for computing a plurality of backup paths that are setup and activated if there is “a failure condition associated with at least one of said working path and one of said backup paths.” Ex.1005, Claim 31; *see also* Ex.1005, [0008]-[0009], [0025]-[0026], [0031]-[0034]; Ex.1003, ¶72.

In “exemplary” FIGS. 5A-5E, EDC_525 discloses computing and setting up a working path 502 and disjoint backup path 504 (first protection path) and backup path 506 (second protection path) that provide protection in case of multiple failures. Ex.1005, [0032], [0045], FIGS. 5A-5E; Ex.1003, ¶73.



Ex.1005, FIGS. 5A to 5E.

EDC_525's instructions for computing uses "network topology" information from a global database to compute backup paths that are completely or partially disjoint. Ex.1005, [0026]-[0032]. After the working and at least one backup path is computed, "appropriate setup and/or activation messages" are transmitted to the other nodes. Ex.1005, [0031]; *see also* Ex.1005, [0008], FIG. 4; Ex.1003, ¶73.

EDC_525 further discloses that one or more backup paths may be dynamically computed using a suitable backup path technique. Ex.1005, [0042]. In one example, a backup path is dynamically computed if a quality parameter is below a certain threshold and in another example the computation is based on a defined Shared Risk Link Group (SRLG) that is predictive of correlated failures.

Ex.1005, [0042], Fig. 10; *see also* Ex.1005, [0008], [0025], [0026], Claim 27;
Ex.1003, ¶74.

EDC_525 provides numerous path computation techniques, including partially and completely disjoint techniques. *See* Ex.1005, FIGS. 4, 6-10 and corresponding text. Ex.1005, [0033], [0037]-[0038], [0040]-[0041], Claims 21, 33, 36, FIGS. 6-10; Ex.1003, ¶¶75-76.

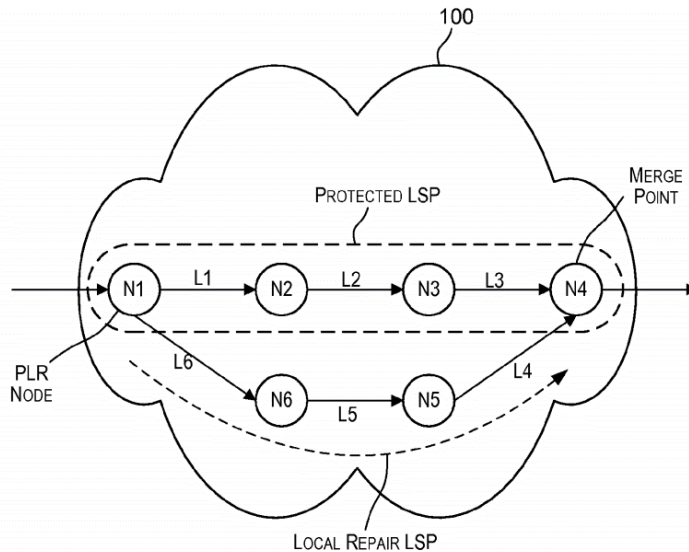
2. EDC_892

EDC_525 expressly refers to the teachings of EDC_892 by its application number (09/998,362) and incorporates EDC_892's contents by reference. *See e.g.*, Ex.1005, [0031]; Ex.1003, ¶¶77-78.

EDC_525 discloses the use of a global database (also referred to as global allocation database) and the use of “protection messages” for updating the global allocation database of each node. Ex.1005, [0028]-[0031]. EDC_892 explains that the “protection message” may be a setup message with a field that identifies the connection as used for protection and a “field that identifies the working path that needs the protection.” Ex.1006, [0033]; *see also* Ex.1006, [0036]. Additionally, EDC_892 further describes computing backup paths. *See e.g.*, Ex.1006, [0020]-[0040], FIGS. 2-4; Ex.1003, ¶¶79-80.

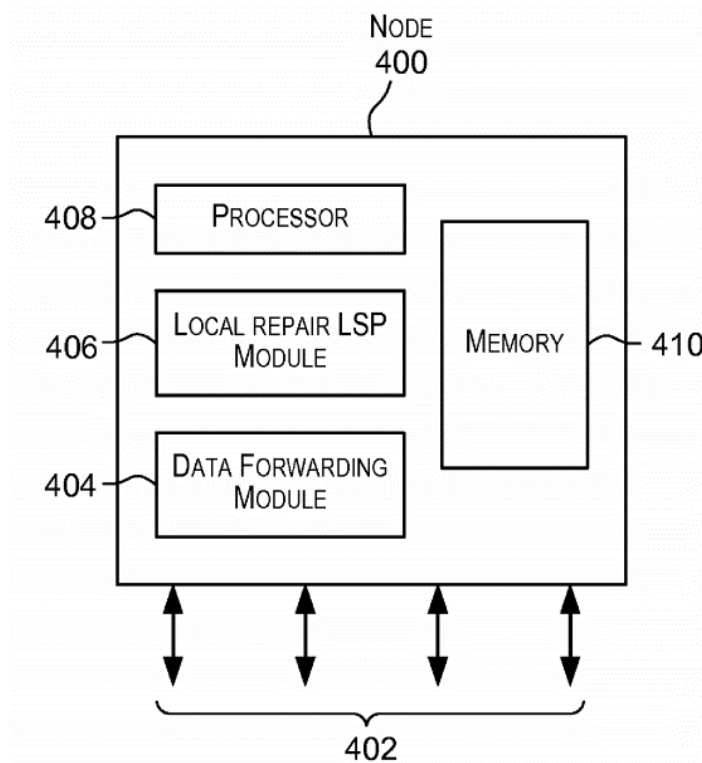
3. Hanif

Hanif generally pertains to local repair connections in an MPLS network, illustrated below at FIG. 1. Ex.1013, [0002], [0028], Abstract; Ex.1003, ¶¶81-82.



Ex.1013, FIG. 1.

Hanif's illustrated MPLS network includes an ingress point of local repair node N1 ("PLR NODE") and an egress node N4 ("MERGE POINT") where the protected LSP merges with the local repair LSP. Ex.1013, [0032]-[0037]; Ex.1003, ¶83. Hanif illustrates in FIG. 4 a network node, which may be implemented as a router or switch with memory for storing network topology information and a processor that executes software to perform the various functions, including determining local repair LSPs based on the network topology information. Ex.1013, [0066], [0069], [0072]-[0074]; Ex.1003, ¶84.



Ex.1013, FIG. 4.

Hanif also teaches outputting an error condition when a local repair LSP cannot be established. Ex.1013, [0050]; Ex.1003, ¶85.

4. Reasons to Combine EDC_892 with EDC_525.

a. EDC 525 and EDC 892 are Analogous Art

EDC_525 and EDC_892 are both analogous art because they pertain to computer networks that use backup paths, like the '691 patent. Ex.1001, 1:6-10, Abstract; Ex.1005, [0001]- [0009], Abstract, FIGS. 4, 6-9, Claims 1, 31; Ex.1006, [0031], Abstract. Additionally, both EDC_525 and EDC_892, like the '691 patent, address the additional problems of using network topology information, SRLG

information, or shortest path first information to calculate backup paths. Ex.1001, 2:10-13, 3:1-25; Ex.1005, [0022], [0026]-[0030], [0042], Claim 32; Ex.1006, [0012], [0025]-[0033], [0038], [0047]; Ex.1003, ¶86.

b. Motivation to combine EDC 892 with EDC 525

A POSITA would have considered and combined the teachings because EDC_525 suggests the combination by expressly citing to EDC_892 and incorporating it by reference. Ex.1005, [0031]; Ex.1003, ¶87.

EDC_525 discloses that once backup path computations are completed, appropriate setup and/or activation messages are transmitted from the nodes along the path in the network. Ex.1005, [0031]. Consistent with EDC_525's disclosure (*see* Ex.1005, [0030]), EDC_892 teaches that a "SETUP" message (1) indicates that the connection is for a protection path and (2) identifies the working path that needs the protection path. Ex.1006, [0033]; Ex.1003, ¶88.

It would have been obvious to a POSITA to include in EDC_525's setup message an indication that the computed connection is used for protection and also identify the working path that needs the protection, per EDC_892, because such information would be useful "for updating the global allocation database" of the receiving nodes. Ex.1006, [0033]. This is consistent with EDC_525's disclosure that a node's "global allocation database may be compiled from allocation

information provided by other nodes.” Ex.1005, [0030]; *see also* Ex.1005, [0028]-[0030]; Ex.1003, ¶89.

The proposed combination of EDC_892 with EDC_525 is nothing more than combining prior art elements (e.g., a setup message that specifically identifies the working path that is protected by a given backup path, per EDC_892, with EDC_525’s setup messages) according to known methods to yield predictable results (e.g., providing information for updating a node’s global database). Ex.1003, ¶90.

c. Expectation of success

The results would have been predictable and there would have been a reasonable expectation of success (1) since EDC_525 incorporates EDC_892’s teachings into its own specification and (2) given the similarities of the two references. Additionally, a POSITA would have known how to use well-known software, hardware, and signaling techniques to implement the proposed combination. Ex.1003, ¶91.

5. Reasons to Combine Hanif with EDC_525

a. Hanif is Analogous Art

Hanif discloses providing a backup path (referred to as a local repair LSP) for a protected path in an MPLS network and is therefore analogous art to the ’691 patent. Ex.1001, 1:6-10, Abstract; Ex.1013, [0002], [0012], [0028], Abstract.

Additionally, like the '691 patent, Hanif addresses the problem of using a processor to perform a method for providing the backup path in the MPLS network. Ex.1001, 2:14-28, 6:4-50, FIG. 4; Ex.1013, [0015], [0066]-[0067], [0069], [0072], [0074], FIG. 4; Ex.1003, ¶92.

b. Motivation to combine Hanif with EDC 525

A POSITA would have been motivated to combine the teachings of Hanif and EDC_525 (as modified in view of EDC_892) to produce numerous predictable and beneficial results. Ex.1003, ¶93.

i. a network processor of a Multiprotocol Label Switching (MPLS) label switch router

EDC_525 discloses that its network node has a “processor-accessible medium...[with] instructions for carrying out network operations.” Ex.1005, Claim 31. EDC_525, however, provides limited details regarding how the instructions are used to carry out the network operations. It was well-known in the art—indeed conventional—for the instructions to be executed by a processor. Ex.1003, ¶¶49, 69, 94; *see also* Material Facts #11; Ex.1008, 20:41-43; Ex.1011, 2:12-15; Ex.1012, 3:31-33.

Hanif discloses that instructions are “executed by [a] processor.” Ex.1013, [0074]; *see also* Ex.1013, [0060],[0073]. It would have been obvious to a POSITA to consider and apply the teachings of Hanif, when implementing EDC_525’s

network node, and execute the instructions stored on the processor-accessible medium to achieve the results that EDC_525 is already describing; namely, utilizing the instructions to carry out network operations. Ex.1003, ¶95.

The combination of Hanif with EDC_525 merely represents a simple combination of known elements (e.g., Hanif’s processor that executes instructions with EDC_525’s network node that includes instructions accessible by a processor) to yield predictable results (e.g., enabling EDC_525’s network node to execute the instructions to carry out network operations). Ex.1003, ¶96.

ii. a Point of Local Repair node and a Merge Point node

EDC_525 discloses that its optical network has “a working path between an ingress node and an egress node.” Ex.1005, [0008]. EDC_525 also contemplates that other devices may be connected before the ingress node and beyond the egress node, as illustrated at FIG. 1. Ex.1003, ¶97.

It was recognized in the art as important for repair to be localized within a network because local repair allows for addressing failures more quickly, efficiently, and avoiding or reducing effects of failures on other adjacent networks, among other benefits. Ex.1003, ¶¶44, 98; *see also* Material Facts #9; Ex.1011, 1:38-45; Ex.1023, 278, 281; Ex.1022, 295. To that end, Hanif discloses local protection where an ingress node is implemented as a point of local repair and an

egress node is implemented as a merge point where the protected path and the repair path merge. Ex.1013, [0007]; *see also* Ex.1013, [0002]-[0009], [0012], [0035]-[0039], [0060]-[0062], Claim 2, Claim 20, FIGS. 1, 2, 3; Ex.1003, ¶99.

It would have been obvious to a POSITA to consider and apply Hanif's local repair teachings, when implementing EDC_525's teachings such that the ingress node is a point of local repair and the egress node is a merge point, to facilitate repairing a failed path quickly and efficiently and to avoid or reduce effects of failures to other adjacent networks, among other benefits. Ex.1003, ¶¶44, 98, 100; *see also* Material Facts #9. The noted benefits, separately and together would have motivated a POSITA to make the proposed combination. Ex.1003, ¶100.

The proposed combination merely represents the application of a known technique (e.g., Hanif's technique of implementing the ingress node as point of local repair and an egress node as a merge point, to EDC_525's ingress and egress nodes) to yield predictable and beneficial results (e.g., quickly and efficiently repair failures and avoid or reduce effects of failures on other adjacent networks, among other benefits). Ex.1003, ¶101.

c. Expectation of success

The results would have been predictable and there would have been a reasonable expectation of success in the combination given the similarities in EDC_525 and Hanif, as analyzed above in prior art summary section. Also, the

results would have been predictable and there would have been a reasonable expectation of success in the combination since processors, as disclosed by Hanif, were components well-known in the art and specifically designed to execute instructions, as EDC_525 discloses. *See* Material Facts 11-12; Ex.1003, ¶102. Moreover, there would have been a reasonable expectation of success in implementing EDC_525's ingress node and egress nodes as a local repair node and a merge node, respectively, as evidenced by Hanif itself and because local repair was well known. *See* Material Facts 8-10; Ex.1009, 144; Ex.0013, [0007]-[0008]. Accordingly, a POSITA would have possessed the skills required to make the proposed combination with a reasonable expectation of success. Ex.1003, ¶102.

Additional analysis supporting the obviousness of the combination of EDC_525, EDC_892, and Hanif is provided in the detailed claim analysis below. Ex.1003, ¶103.

6. Claim 1

EDC_525 in combination with EDC_892 and Hanif renders obvious Claim 1. Ex.1003, ¶67.

[1.0.1] *A method performed by a network processor of a Multiprotocol Label Switching (MPLS) label switch router*

First, EDC_525 discloses a “*method*,” by teaching a “**method** for implementing a shared protection scheme.” Ex.1005, [0008]; *see also* Ex.1005,

FIGS. 4, 6-9, Claims 1, 31. Additionally, EDC_525 expressly refers to and incorporates EDC_892's contents by reference. Ex.1005, [0031]. EDC_892 also discloses "**a method**...for providing shared path protection." Ex.1006, [0004]; *see also* [0009], Claim 1. It would have been obvious to a POSITA considering EDC_525 to also refer to and apply EDC_892's teachings because EDC_525 expressly directs and encourages this. *See also* Reasons to Combine EDC_892 with EDC_525; Ex.1003, ¶¶104-105.

EDC_525 renders obvious that its method is "*performed by a network processor of a Multiprotocol Label Switching (MPLS) label switch router.*" In one example, EDC_525's method is implemented in a "**generalized multi-protocol label switched (GMPLS)** optical transport network," illustrated below at FIG. 1. Ex.1005, [0022], FIG. 1.

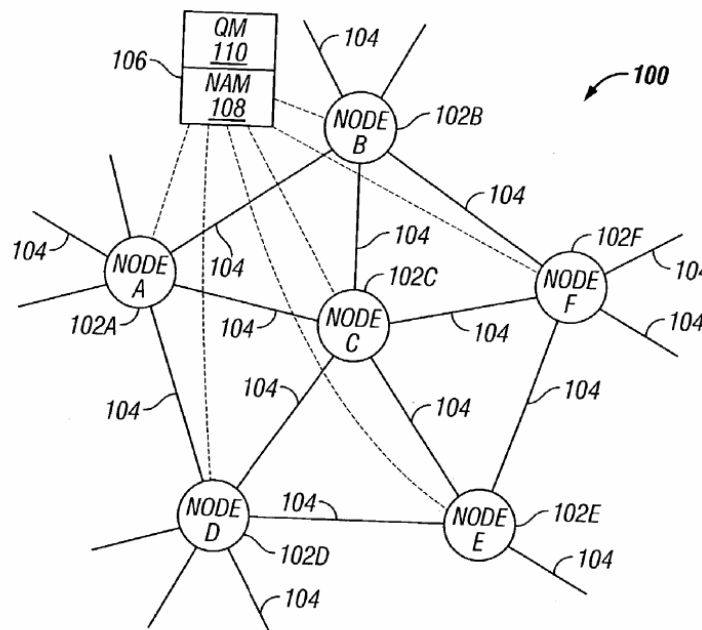


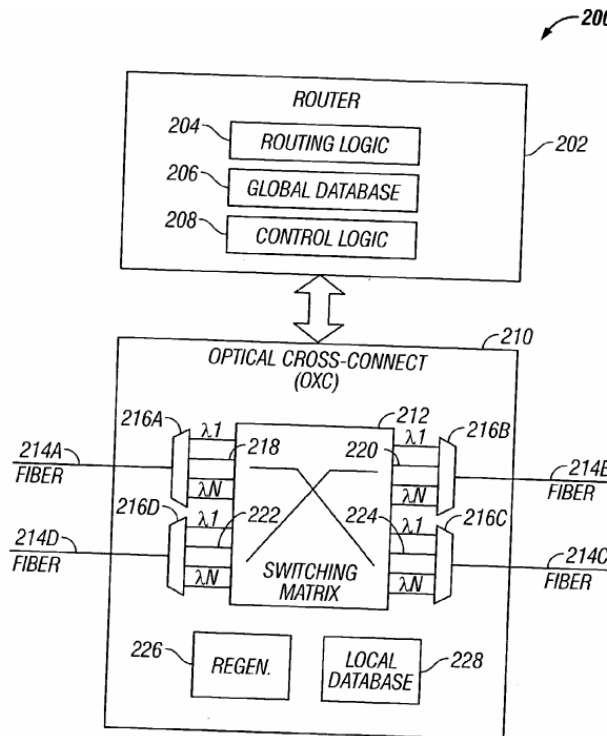
FIG. 1

Ex.1005, FIG. 1.

A POSITA would have recognized that GMPLS is a version of MPLS designed for optical networks, among others. *See e.g.*, Ex.1003, ¶¶33-34, 69, 107; *see also* Material Facts #1-2; Ex.1011, 1:25-27; Ex.1018, 1. Accordingly, EDC_525's GMPLS network renders obvious "*Multiprotocol Label Switching (MPLS)*." Ex.1003, ¶¶106-108.

EDC_525 discloses a "*label switch router*" by teaching that the GMPLS network includes a network administrator manager ("NAM") and a quality monitor ("QM") that compute paths and that these elements "may be disposed centrally or distributed over one or more nodes." Ex.1005, [0025]; *see also* Ex.1005, [0042]. EDC_525's network node includes a "**routing part...routing protocol**

logic...and control logic” that performs routing, switching with optical cross-connect (“OXC”), and also “control[s] signaling in the network.” Ex.1005, [0026]-[0027], FIGS. 1, 2. EDC_525’s routing network node 200 is illustrated below.
Ex.1003, ¶109.



Ex.1005, FIG. 2.

EDC_525’s GMPLS network node, including its router and switching functionality, renders obvious a “*label switch router*.” Ex.1003, ¶110.

EDC_525 renders obvious a “*network processor*” by teaching that its network node has “**processor**-accessible medium” and a “**control structure**” that “may be **embodied in software, hardware, or any combination thereof**.”

Ex.1005, [0009], Claim 31; *see also* Ex.1005, [0025]-[0026], [0031], [0042].

A POSITA would have found it obvious to implement the control structure in hardware with a “*processor*” because EDC’525 describes a “processor-accessible medium” with executable “instructions for carrying out network operations.” *See e.g.*, Ex.1005, Claim 31. Such an implementation would have been consistent with well-known and commonly utilized techniques in the art. Ex.1003, ¶¶49, 69, 94, 111-112; *see also* Material Facts #10-11; Ex.1008, 20:41-43; Ex.1011, 2:12-15; Ex.1012, 3:31-33.

Second, to the extent argued that “*a network processor of a Multiprotocol Label Switching (MPLS) label switch router*” is insufficiently disclosed above, the further combination with Hanif renders it obvious. Ex.1003, ¶113.

Hanif discloses that a method to setup local repair paths is performed by a node “**embodied as a network device such as a switch or router” in a “Multi-Protocol Label Switching (MPLS)” network**. Ex.1013, [0028]-[0030], [0040], [0066]. The node utilizes a “**processor 408 or [] processor within module 406**” to perform network operations, such as determining paths and forwarding packets, among other network operations. Ex.1013, [0015], [0039], [0061]-[0069], [0072], [0074], FIG. 4. Forwarding is performed by “switching fabric” using a “label switching protocol.” Ex.1013, [0004], [0015], [0069], Claim 3. Hanif’s processor 408 and the processor within module 406 of the node, separately and together, render obvious “*a network processor of a Multiprotocol Label Switching (MPLS)*

label switch router.” Ex.1003, ¶114.

It would have been obvious to a POSITA to apply Hanif’s teachings to EDC_525 and implement the control structure hardware of each network node with a processor, because a processor is specifically designed to access EDC_525’s “processor-accessible medium” and execute the stored “instructions for carrying out the network operations.” Ex.1005, [0009], Claim 31; Ex.1013, [0069], [0072]. A POSITA would have recognized that using a processor to execute the instructions to carry out the network operations would facilitate the computations of multiple backup paths and other network operations, thereby furthering EDC_525’s objectives. *See e.g.*, Ex.1005, [0034]; *see also* Material Facts #10-11; Reasons to Combine Hanif with EDC_525; Ex.1003, ¶115.

Moreover, to the extent argued that GMPLS optical networks are outside the scope of the ’691 patent, a POSITA would have recognized that the disclosure of EDC_525 and EDC_892 is relevant to other types of networks; they are not limited to optical networks. Ex.1005, [0003], [0045], Claim 1 (not reciting any optical limitation); Ex.1003, ¶116. That is, a POSITA would have recognized that the combined teachings are applicable when implementing IP networks that utilize MPLS, as disclosed by Hanif. Accordingly, it would have been obvious to a POSITA to apply the combined teachings of EDC_525, EDC_892, and Hanif when implementing IP networks that utilize MPLS to obtain the predictable results of

using local protection (per Hanif) while addressing multiple simultaneous network failures (per EDC_525 and EDC_892). Ex.1003, ¶116.

Thus, to the extent the preamble is limiting, EDC_525 alone and in combination with EDC_892 and Hanif discloses a method performed by a network processor of a MPLS label switch node (e.g., implemented as a router), which renders it obvious. Ex.1003, ¶117.

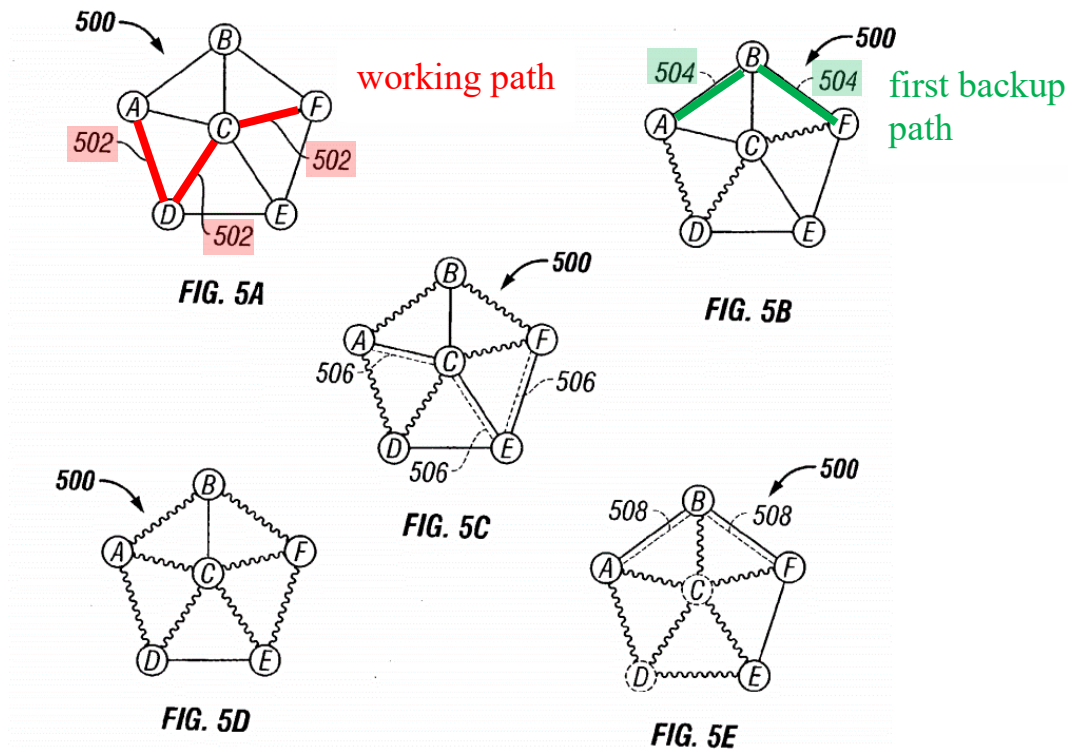
[1.0.2] for providing a Backup Label Switched Path (LSP) to a Bypass LSP already established for a Protected Primary LSP, the method comprising the steps of:

First, EDC_525 discloses “a Bypass LSP already established for a Protected Primary LSP.” EDC_525 teaches that its “method commences by computing a **working path**” and that a “**first protection path**^[4] is computed using any known algorithm.” Ex.1005, [0008]-[0009], [0031]-[0034]; *see also* analysis at [1.1], *infra*; Ex.1003, ¶¶118-119.

In FIGS. 5A and 5B below, an “exemplary working path [is] denoted by reference numeral 502” and the first protection or backup path corresponds to

⁴ EDC_525 uses the terms “protection path” and “backup path” interchangeably, consistent with the art. *See, e.g.*, Ex.1005, [0030], [0034]; Ex.1022, 296. Ex.1003, ¶119 n.1; Material Facts #16.

“protection path 504.” Ex.1005, [0032], FIGS. 5A, 5B; *see also* Ex.1005, Abstract; Ex.1003, ¶120.



Ex.1005, FIGS. 5A-5E (annotated).

These figures are merely “exemplary” (Ex.1005, [0032]) and a POSITA would have understood that EDC_525’s teachings apply generally to other network topologies that may comprise a lesser or greater number of nodes or links. *See* Ex.1005, [0045], FIG. 1 (illustrating a broader network), FIG. 3A (disclosing additional path Q,R,S); Ex.1003, ¶121. In other network topologies, for example, a POSITA would have understood that the working path and the first backup path

may be different. Ex.1003, ¶121. Also, a POSITA would have recognized that the path calculations may be performed in a different sequence in different network conditions (e.g., based on link availability and load) such that a different working path and a different first backup path are established. Ex.1003, ¶121. Additionally, EDC_525 teaches calculating “one or more backup paths” and “a predetermined number” of backup paths. Ex.1005, [0041]-[0042]. As such, a POSITA would have understood that EDC_525’s disclosure is open ended and that one or a greater number of backup paths may be initially provided. Ex.1003, ¶121.

EDC_525 renders obvious that the working path and first backup path are “*established*” by teaching that “**setup and/or activation messages**” regarding the working path and backup path “are then transmitted to the nodes spanning the paths.” Ex.1005, [0008], [0031]; *see also* Ex.1006, [0036]-[0047], FIG. 4; Ex.1003, ¶122. EDC_525’s setup signaling is consistent with well-known techniques to “*establish[]*” paths. Ex.1003, ¶¶35-36, 123; *see also* Material Facts #3-4; Ex.1011, 3:7-9, 4:7-9; Ex.1018, 37. It also would have been obvious for the working and first backup paths to be “*established*” in view of EDC_525’s disclosure of monitoring quality along paths during operation. *See, e.g.*, Ex.1005, [0008]; Ex.1021, 2:1-3; Ex.1003, ¶123.

Moreover, it would have been obvious to a POSITA to “*establish[]*” the first protection path for the working path before operation begins, because it was

recognized in the art that “**preestablishment of protection resources is fundamental for any protection strategy**” and that establishing a protection path after a failure is “too late.” Ex.1022, 293; Ex.1003, ¶124.

Accordingly, EDC_525’s disclosure of transmitting setup and/or activation messages for a first backup path and a working path renders obvious “*a Bypass LSP already established for a Protected Primary LSP.*” Ex.1003, ¶125.

Additionally, EDC_892 discloses that “**setup messages are sent to the working path and protection path nodes**,” for “a new protection path to protect a defined working path.” Ex.1006, [0010], [0036], Claim 1; *see also* Ex.1006, [0033], Claims 5, 15. That EDC_892’s setup messages establish the backup path is confirmed by its cited provisional application, which discloses that the “**protection path is established just after the calculation of the working path.**” Ex.1007, 2. It would have been obvious to combine the teachings of EDC_892 with EDC_525. *See* Reasons to Combine EDC_892 with EDC_525. Thus, EDC_525 in combination with EDC_892 also renders obvious “*a Bypass LSP already established for a Protected Primary LSP.*” Ex.1003, ¶126.

Second, EDC_525 discloses “*providing a Backup Label Switched Path (LSP)*” by teaching that “**another iteration of a protection path computation takes place**” to provide a “**second protection path**.” Ex.1005, [0032], FIGS. 5A, 5B, 5C; *see also* Ex.1005, Claim 31; Ex.1003, ¶127.

EDC_525 further discloses that the “backup path computation...may be...dynamically invoked based on network quality.” Ex.1005, [0042]. For example, if a quality parameter is below a certain threshold (indicative of a “degradation event”) the backup path computation is dynamically invoked to provide an additional “**one or more backup paths.**” Ex.1005, [0042], Fig. 10; *see also* Ex.1005, [0008], [0025], [0026]; Ex.1003, ¶128.

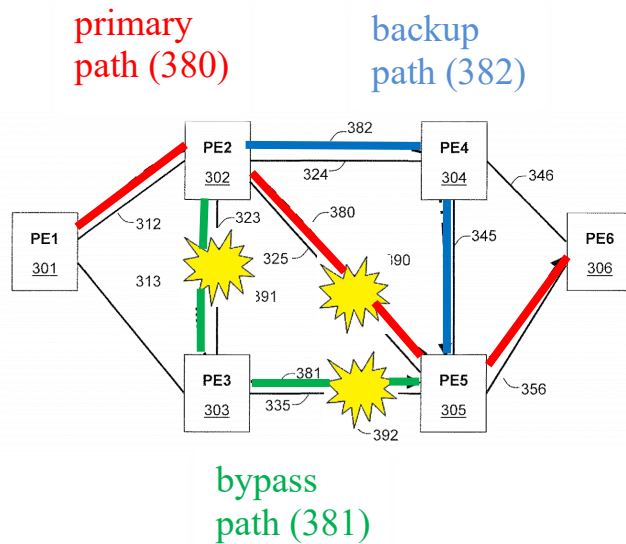
A POSITA seeking to implement the teachings of EDC_525 would have found it obvious to monitor network quality, after establishing the working and first backup paths, and to dynamically provide an additional “one or more backup paths” in case the established working path and first backup path are observed to have diminished quality. Ex.1003, ¶129. Of note is that this disclosure in EDC_525 relates to establishing an additional (second) backup path while the working and first backup paths remain operational—before failure. Thus, this disclosure is unlike the prior art distinguished during prosecution, where an additional backup path was established only upon a failure of the working or first backup path. *See* Ex.1002, 41. The dynamically provided “one or more backup paths” (a second backup path) would further EDC_525’s goal providing “a diverse set of backup paths [that] can provide better protection against multiple failure events.” Ex.1005, [0034]; *see also* Ex.1005, [0031], Abstract; Ex.1003, ¶129.

Moreover, just like the '691 patent's disclosure of considering SRLG information (Ex.1001, 2:44-50), EDC_525 provides a further embodiment that defines an SRLG for early detection of potential failures and discloses providing an additional backup path that is not part of that group. Ex.1005, [0042]. A POSITA would have understood that defining a SRLG is predictive (e.g., identifies paths that may fail together) and that actual failure of a working path or its protection path has not occurred. That is, EDC_525 provides a means for predicting paths that have a risk of concurrent or correlated failures, without actual failure observed. Ex.1005, [0007], [0042]. In circumstances where there is a potential for correlated failures, an additional one or more backup paths would beneficially be computed and setup so as to not be part of the SRLG in case of correlated failure. Ex.1005, [0034], [0042]. Accordingly, it would have been obvious to a POSITA, after establishing the working and first backup paths (as discussed immediately above) to define a SRLG and setup a second backup path that is not part of the SRLG of the working and first backup paths, to thereby "reduce the possibility of multiple failures." Ex.1005, [0042]; Ex.1003, ¶¶130-131.

Additionally, Hanif teaches that "where the traffic needs to be redirected onto a backup or detour tunnel within a specified time limit (e.g., for voice over IP applications), the computing and signaling for the local repair connections is typically done in advance of the failure." Ex.1013, [0009]; *see also* Ex.1013,

[0039]. Hanif also contemplates that there is “at least one local repair connection.” Ex.1013, [0010]. Accordingly, in view of Hanif, it would have been obvious to signal setup of EDC_525’s second backup path before actual failure so that repair is performed quickly in case of correlated or simultaneous failures of both the established working and first backup paths (e.g., because they are part of the same SRLG). Ex.1005, [0042], Claim 15; Ex.1022, 293; Ex.1003, ¶132.

In summary, just like the ’691 patent’s embodiment of FIGS. 2-3, where two backup paths are provided, the prior art backup paths (e.g., a first backup path and a second backup path) provide protection against multiple failures. *See also* Ex.1001, 5:48-58; Ex.1005, [0034]. Below is a side-by-side comparison showing how in both the ’691 patent’s FIG. 3 and EDC_525’s FIG. 5C, the second backup path provides protection in case the first backup path fails simultaneously with the working path. Ex.1003, ¶¶133-134.



Ex.1001, FIG. 3
(annotated).

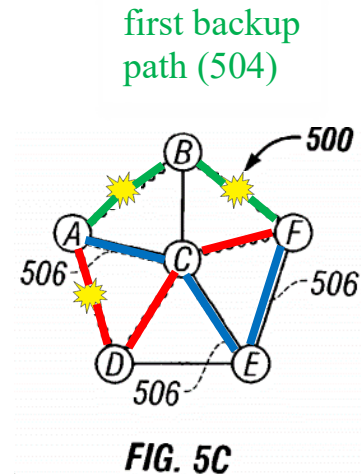


FIG. 5C

working path (502) **second backup path (506)**

Ex.1005, FIG. 5C
(annotated and modified
to show exemplary failure
condition).

Accordingly, EDC_525's second backup path (which is dynamically provided after the working and first backup paths are established) corresponds to the claimed "*Backup Label Switched Path (LSP)*." See also Ex.1005, [0032]-[0042], FIGS. 5 to 10; analysis at [1.6], *infra*; Ex.1003, ¶135.

Thus, to the extent this part of the preamble is limiting, EDC_525 alone and in combination with EDC_892 and Hanif renders it obvious. Ex.1003, ¶136.

[1.1] protecting the Primary LSP against dual failures, comprising:

First, as discussed at [1.0.2], EDC_525 alone and in combination with EDC_892 and Hanif discloses that the method establishes a working path, which

corresponds to “*the Primary LSP*,” and further teaches a first and a second backup path. Ex.1003, ¶¶137-138.

“protecting the Primary LSP against dual failures”

Second, and consistent with [1.0.2], EDC_525 discloses “*protecting the Primary LSP against dual failures*,” by teaching that the working path is “**protect[ed] against multiple failures**” because the “**working path [is] being protected by two backup paths**.” Ex.1005, [0031], [0034]; *see also* Ex.1005, [0008], [0028]-[0042], [0045], FIGS. 6-9; Material Fact #5; Ex.1003, ¶139.

Thus, EDC_525 alone and in combination with EDC_892 and Hanif discloses protecting the working path against multiple failures, which renders obvious this element. Ex.1003, ¶140.

[1.2] *establishing the Bypass LSP for the Protected Primary LSP having a Point of Local Repair node and a Merge Point node*

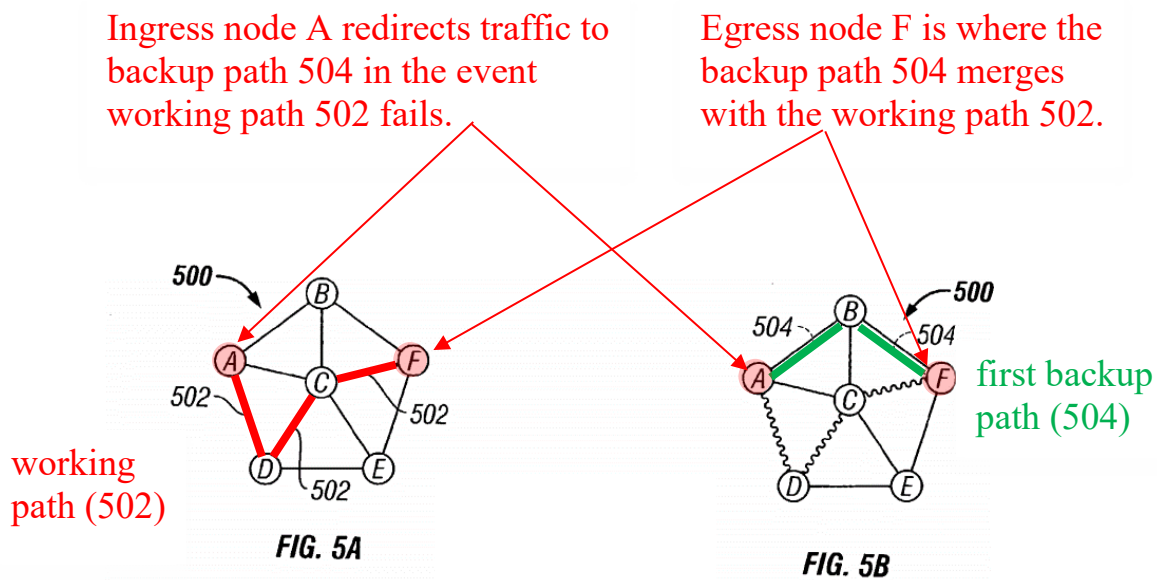
First, as discussed at [1.0.2], EDC_525 alone and in combination with EDC_892 and Hanif discloses that the method includes transmitting setup and/or activation messages to establish the first backup path for the working path, which renders obvious “*establishing the Bypass LSP for the Protected Primary LSP*.” Ex.1003, ¶¶141-142.

Second, EDC_525 discloses that the primary path “*ha[s] a Point of Local Repair node and a Merge Point node*” by teaching that the working path is setup

between “an **ingress node and an egress node.**” Ex.1005, [0008], [0031]; *see also* Ex.1005, [0036]; Ex.1006, [0030], [0036]-[0037], FIG. 4. Ex.1003, ¶143.

In EDC_525’s example of FIGS. 5A and 5B, the working path is denoted by reference numeral 502 and the first backup path has reference numeral 504.

Ex.1005, [0032]; Ex.1003, ¶144.



Ex.1005, FIGS. 5A and 5B (annotated).

It would have been obvious to a POSITA that in the event of working path 502 failure, ingress Node A redirects traffic onto the first backup path 504 that merges with the working path at egress Node F. Ex.1003, ¶145.

A POSITA would have recognized, consistent with knowledge in the art, that the nodes in FIGS. 5A-5E are local to the optical network and that additional path connections may extend to other nodes of other domains. *See* Ex.1005, [0026]

(discussing maintaining information regarding each link in the network domain).

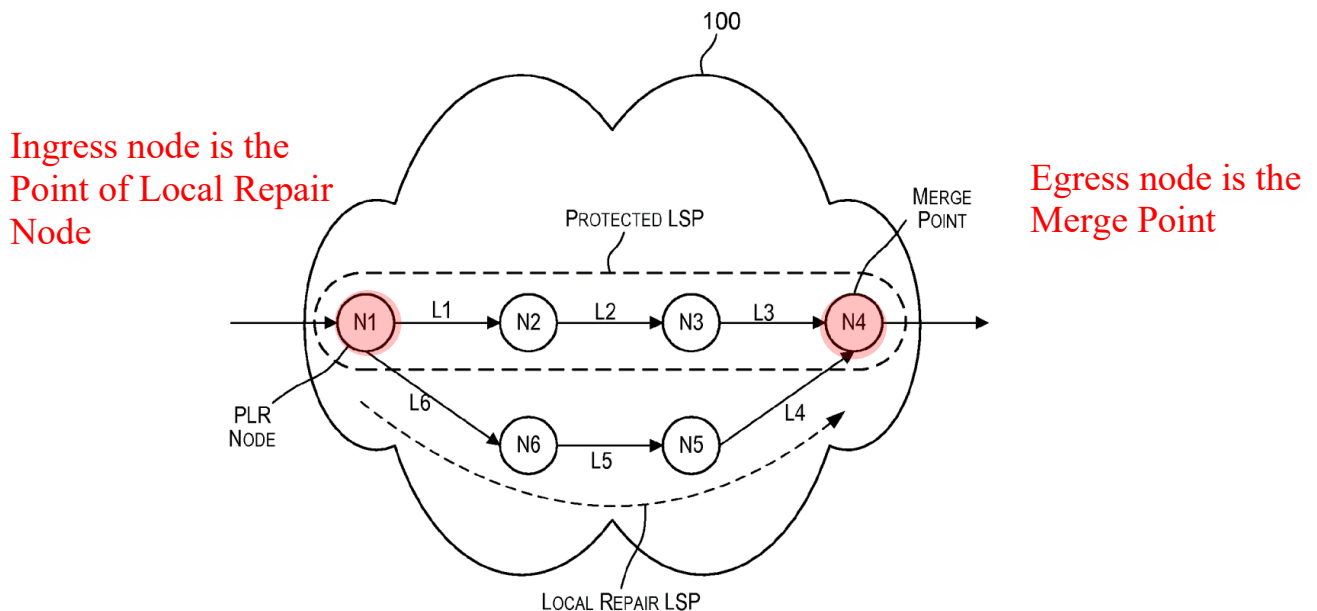
For example, EDC_525's FIG. 1 illustrates that additional paths extend from Nodes A, B, D, E, F. Ex.1005, FIG. 1; *see also* Ex.1006, FIG. 1; Ex.1008, FIG. 1; Ex.1015, Abstract, FIG. 1; Ex.1003, ¶145.

Additionally, a POSITA would have recognized that EDC_525's disclosure is in the context of local protection, at least because it supports sharing of backup paths for scalability. Ex.1003, ¶¶38-42, 146; *see also* Material Facts #6-7; Ex.1005, [0008]; Ex.1022, 295. A global protection scheme, in contrast to local protection, does not permit sharing. Ex.1003, ¶¶38-42, 146; *see also* Material Facts #6-7; Ex.1022, 294. Accordingly, it would have been obvious to a POSITA for EDC_525's nodes to be implemented as "*Local*" nodes, at least because EDC_525 supports shared protection. Ex.1003, ¶146.

Thus, and consistent with the '691 patent, EDC_525's ingress node, which redirects traffic (e.g., along the first backup path 504), corresponds to the claimed "*Point of Local Repair node*." Ex.1001, 1:42-46. Additionally, and consistent with the '691 patent, EDC_525's egress node, where the first backup path merges with the working path, corresponds to the claimed "*Merge Point node*." Ex.1001, 1:42-46; Material Facts #10; Ex.1003, ¶147.

Third, to the extent argued that “a Point of Local Repair node and a Merge Point node ...” is not sufficiently disclosed, the further combination with Hanif renders such obvious. Ex.1003, ¶148.

As discussed immediately above, EDC_525 discloses an “**ingress node**” where traffic is redirected and an “**egress node**” where traffic merges. Ex.1005, [0008]. Hanif discloses “local repair,” where both the protected LSP and the backup LSP share an “**ingress node**” that corresponds to a point of local repair node N1 (“**PLR NODE**”) and an “**egress node**” that corresponds to a merge point node N4 (“**MERGE POINT**”). Ex.1013, [0012], [0032], [0037], Abstract, FIG. 1; *see also* Ex.1013, [0007]-[0009], [0012], [0035]-[0039], [0060]-[0062], Claim 2, Claim 20, FIGS. 1, 2, 3; Ex.1003, ¶149.



Ex.1013, FIG. 1 (annotated).

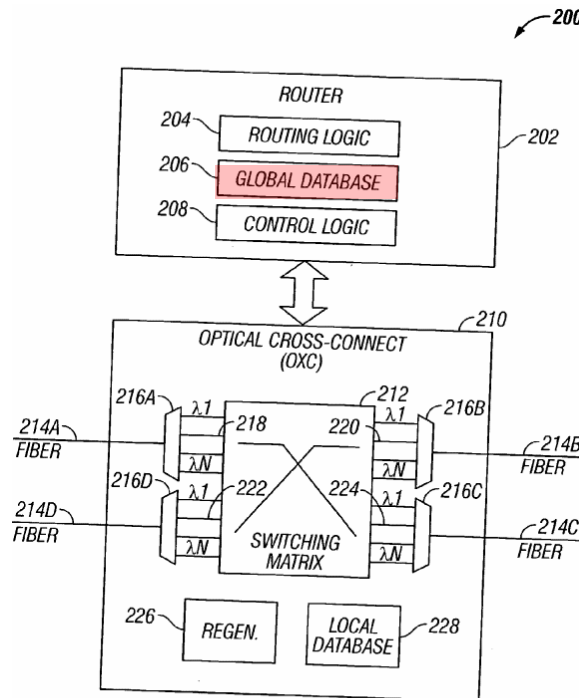
It would have been obvious to a POSITA, in view of Hanif’s local protection teachings, to implement EDC_525’s ingress node as a PLR Node and the egress node as a Merge Point Node. Implementing EDC_525 with local protection, per Hanif, would have been recognized as beneficial because it would allow for repair to be performed (1) quickly and efficiently (2) avoid or reduce effects of failures in the optical network on adjacent networks (e.g., in the instance where the egress node connects to an IP network), (3) consume less network state, and (4) reduce traffic disruption during failure, among other benefits. Ex.1003, ¶¶44, 150; *see also* Material Facts #9; Ex.1009, 144; Ex.1013, [0032]; Ex.1011, 1:38-50; Ex.1022, 295, 340; Ex.1023, 278; Reasons to Combine Hanif with EDC_525.

Thus, EDC_525 in combination with EDC_892 and Hanif discloses establishing a first backup path for the working path having an ingress PLR node and an egress Merge Point node, which renders obvious this element. Ex.1003, ¶151.

[1.3] *obtaining the nodes traversed by an end-to-end path of said Bypass LSP from said Point of Local Repair Node to said Merge Point node;*

First, as discussed at [1.2], a first backup path corresponds to the “*Bypass LSP*,” an ingress node corresponds to the “*Point of Local Repair Node*,” and an egress node corresponds to the “*Merge Point node*.” Ex.1003, ¶¶152-153.

Second, EDC_525 teaches “obtaining the nodes traversed by an end-to-end path” by disclosing that each network node has a “global database” with “**current topology...for intelligent, dynamic creation of network paths under control of control logic**” and that the topology information is for “**each channel of each link in the entire network domain.**” Ex.1005, [0026]. “Information regarding other links in the global allocation database may be **compiled from allocation information provided by other nodes** in the network domain. In one implementation, such information may be shared among all nodes in the network.” Ex.1005, [0030], FIGS. 2, 3A, 3B; Ex.1003, ¶¶154-156.



Ex.1005, FIG. 2 (annotated).

In one example, EDC_525 discloses that network topology information is used to determine backup paths that are both link and node disjoint. Ex.1005, [0008], [0032]-[0033], Claim 6, FIG. 5A-5E. For example, path calculation logically removes from the topology information an already-used path node and then calculates “one or more backup paths between the ETE [end-to-end] nodes.” Ex.1005, [0032]-[0033], [0038]. A POSITA would have understood that in order to calculate backup paths that are node disjoint, the topology information identifies the nodes traversed from the ingress node to the egress node (i.e., end-to-end); these nodes are then avoided when calculating disjoint backup paths. Accordingly, it would have been obvious to a POSITA for the network topology information, which is obtained from “all nodes” for “each link in the entire network domain,” to include information regarding nodes traversed by an end-to-end path of the first backup path (“*Bypass LSP*”) from an ingress node (“*Point of Local Repair Node*”) to an egress node (“*Merge Point node*”). Ex.1005, [0030], [0026]; Ex.1003, ¶157.

Additionally, EDC_892 discloses a “global allocation database 26 [that] comprises one or more tables that provide **a current topology of the network 10 for intelligent, dynamic creation of network paths,**” where a path corresponds to “a sequence of **nodes**.” Ex.1006, [0023], [0025]. EDC_892’s database “provides information regarding each channel of each link in the network domain.” Ex.1006, [0026]; *see also* Ex.1006, [0032]. As analyzed above, it would have been obvious

to combine the teachings of EDC_892 with EDC_525. *See* Reasons to Combine EDC_892 with EDC_525. Ex.1003, ¶158.

Third, to the extent argued that “*obtaining the nodes traversed by an end-to-end path*” is not sufficiently disclosed, the further combination with Hanif renders such obvious. Ex.1003, ¶159.

Hanif teaches that its node includes “[p]rocessor 408 [] configured to perform processing for tasks performed by node 400” and a “processor within module 406” that may perform “creation of local repair LSPs, optimization of LPATHs, and the like.” Ex.1013, [0072]-[0073]; *see also* Ex.1013, Claim 19. Hanif’s processors may be implemented separately, together, or processor 408 “may...aid modules 406 and 404 in functions performed by those modules.” Ex.1013, [0073]. In one example, processor 408 accesses memory 410 that includes “**network topology information that is used for determining local paths associated with local repair LSPs.**” Ex.1013, [0074]; Ex.1003, ¶160.

In view of Hanif, it would have been obvious to a POSITA when implementing the combination with EDC_525 (See [1.0.1]) for the processor to obtain the network topology information stored in memory and provide the information to the instructions for computing paths to thereby determine local backup LSPs that are disjoint. Ex.1005, [0026], [0033]-[0041], Abstract, Claims

31 and 34, FIGS. 6-8; Ex.1013, [0073]-[0074]; analysis, *infra*, at [1.4]-[1.6];
Reasons to Combine Hanif with EDC_525; Ex.1003, ¶161.

Thus, EDC_525 alone and in combination with EDC_892 and Hanif discloses obtaining network topology information, including the nodes and links traversed by the end-to-end path of the first backup path from an ingress node to an egress node, which renders obvious this element. Ex.1003, ¶162.

[1.4] *generating a request to a path calculator using the nodes traversed by said end-to-end path of said Bypass LSP for a disjoint path connecting said Point of Local Repair Node to said Merge Point node;*

First, as discussed at [1.2], a first backup path corresponds to the “*Bypass LSP*,” an ingress node corresponds to the “*Point of Local Repair Node*,” and an egress node corresponds to the “*Merge Point node*.” Further, as discussed at [1.3], the global database maintains obtained network topology information that includes the nodes traversed by the end-to-end path of the first backup path (e.g., 504), which corresponds to “*the nodes traversed by said end-to-end path of said Bypass LSP*.” Ex.1003, ¶¶163-164.

Second, EDC_525 teaches “*a path calculator*” by disclosing “**processor-accessible medium having... instructions for computing a plurality of backup paths between said ingress and egress nodes.**” Ex.1005, Claim 31; *see also* FIGS. 6-10 (disclosing backup path computation steps). EDC_525’s instructions to compute a plurality of backup paths, taken separately and together with the

processor (See [1.0.1]), corresponds to “*a path calculator*,” as claimed. Ex.1003, ¶165.

Additionally, EDC_525 cites to EDC_892 for “well-known techniques” to calculate paths. Ex.1005, [0031]. In that regard, EDC_892 discloses “calculat[ing] a protection path” and provides various details regarding the calculation, thereby further rendering obvious a “*path calculator*.” Ex.1006, [0036]-[0047], [0052], FIGS. 4, 5. As analyzed above, it would have been obvious to combine the teachings of EDC_892 with EDC_525. *See* Reasons to Combine EDC_892 with EDC_525. Ex.1003, ¶166.

“generating a request...for a disjoint path connecting...”

Third, EDC_525 renders obvious “*generating a request... for a disjoint path connecting said Point of Local Repair Node to said Merge Point node.*” EDC_525’s “instructions for computing a plurality of backup paths” are used to compute a disjoint second backup path that connects the ingress node to the egress node. Ex.1005, [0031]-[0032] (“...a complete link disjoint scheme is used for calculating multiple backup paths.”), [0037], Claims 31, 33, 34, Abstract; Ex.1003, ¶167. And, consistent with [1.0.2], EDC_525 discloses that the second backup path computation may be “dynamically invoked,” e.g., if a quality parameter is below a certain threshold or if a potential concurrent failure is identified (e.g., because of SRLG). Ex.1005, [0042], Fig. 10; *see also* Ex.1005, [0008], [0025], [0026];

Ex.1003, ¶¶167-168. In situations where the working path and first backup path (established based on the connection request, see [1.0.2]) are impacted by the “degradation event” or are part of the SRLG, it would have been obvious to a POSITA to invoke the computing instructions to dynamically calculate an additional backup path with an appropriate computation scheme. *See e.g.* Ex.1005, FIG. 5 (steps 406 to 408), FIG. 6 (steps 608 to 620), FIG. 7 (steps 708 to 720), FIG. 8 (steps 808 to 810), FIG. 9 (steps 908 to 910), FIG. 10 (1008). Such an invocation of backup path computation would beneficially provide additional protection to the working path in case the degradation event gives rise to a failure or the SRLG fails. Ex.1003, ¶168.

Additionally, Hanif teaches that backup path (local repair LSP) calculations “**may be initiated upon receiving a signal.**” Ex.1013, [0042]. Hanif’s backup path calculation considers nodes already used (downstream of the PLR) and does not recalculate the already established protected LSP. Ex.1013, [0042]-[0043], FIG. 2. Accordingly, consistent with the analysis immediately above, it would have been obvious to a POSITA to generate a signal requesting that EDC_525’s computing instructions dynamically calculate an additional second backup path, without recalculating the already established working path and first backup path. Ex.1003, ¶169. Moreover, in one example, Hanif’s backup path calculation is initiated “**whenever a new LSP is provisioned.**” Ex.1013, [0042], FIG. 2. In

instances where EDC_525's "predetermined number of backup paths" is two or more (Ex.1005, [0041]) and only one backup path is initially available, it would have been obvious to a POSITA to initiate backup path calculation whenever a new LSP is provisioned, as Hanif teaches, so that the predetermined number of backup paths is reached. *See also* Reasons to Combine Hanif with EDC_525; Ex.1003, ¶169.

EDC_525 teaches that its "instructions for computing a plurality of backup paths" correspond to computer "**software**" and an "**algorithm**," and may be implemented "**using any technique**." *See, e.g.*, Ex.1005, [0009], [0032]-[0033], Claims 2, 17, 31, 32. It was known in the art to implement software algorithms in a modular fashion using a software subroutines known as functions that would be executed upon request. Ex.1014, 199 ("**function**...A general term for a subroutine."); Ex.1016, 95 ("Most modern [programming] languages have an ability to create named subroutines or subprograms...called a **function**."); Ex.1017, 37 ("The **function is the heart**...[of] program[s]."); Ex.1033, 380 (describing a "**function**" as "a self-contained **software** routine that performs a job for the program it is written in or for some other program."); Ex.1003, ¶170.

A POSITA would have found it obvious to utilize a processor (see [1.0.1]-[1.0.2]) to execute the instructions for computing a plurality of backup paths dynamically, e.g., based on network quality or a defined SRLG. Ex.1005, [0042].

In doing so, it would have been obvious to a POSITA for the processor executing the instructions to generate a request (known in the art as a “function call”) to perform path computation and to provide the obtained network topology information to be used in the path computation (for example, as parameters or arguments). Ex.1003, ¶¶49, 170, 111-112, 170, 179; *see also* Material Facts #11-12; Ex.1012, 3:13-17 (“...control part being configured to **request computation by sending a request to the local path computation element** for computation of the new recovery path....”); Ex.1014, 200 (“**function call...A program’s request...along with any parameters needed for the function** to perform its task.”); Ex.1016, 96 (“[T]he argument list (which follows the name and is surrounded by parentheses) contains the types of **arguments that must be passed to the function.**”); Ex.1017, 37 (“Function arguments are contained in parentheses following the function name. The **values of the arguments are the parameters needed to execute the function.**”); Ex.1033, 380 (“**function call A request by a program...A function call written in a program states the name of the function followed by any values or parameters that have to be passed to it.**”). Ex.1003, ¶171.

171. A POISTA would have known how to implement EDC_525’s network topology information in the global database in numerous ways, including as an array of arrays; in such an implementation, the network topology information

would be passed to the function call, for example, as a pointer to the array of arrays. Ex.1035, 341, 394; Ex.1036, 125, 133; Ex.1037, 11-12, 20; Ex.1038, 321-379. The function would use the pointer to access the network topology information from the arrays and would perform path calculations, including, depending on the desired computation, logically removing used links or nodes, which comports with the claimed “*using*.” Additionally, as another example, a POSITA would have known that the network topology information in the global database may be implemented as a file, and the path-computation function would receive as a parameter the file name (or a pointer to the file). Ex.1034, generally; Ex.1036, 221, 403-404; Ex.1038, 379-392, 651-672; Ex.1039, 1, 10. In such a circumstance, the function would retrieve the network topology information from the named file and perform path calculations, including using the information to logically removing nodes to compute a completely disjoint second backup path. Ex.1003, ¶172.

Accordingly, it would have been obvious to a POSITA for EDC_525’s instructions for computing a plurality of backup paths to be invoked by a request (e.g., function call) that uses the obtained network topology information (e.g., a parameter pointer to an array or file of the information). *See supra* [1.3]; Ex.1005, [0026], [0030]. For instance, in EDC_525’s example of dynamically invoking path computations based on network quality (see Ex.1005, [0042]), it would have been

obvious to a POSITA for a processor to generate a request for a second backup path using a parameter corresponding to the network topology information (which includes nodes traversed by the end-to-end path of the first backup path (see [1.0.2])) because such a request would inform the instructions for computing which nodes should be logically removed from the computations so that the computed backup path is completely disjoint. *See e.g.*, Ex.1005, [0008], Abstract; *see also* Ex.1005, [0031]-[0032], [0041]-[0044], Claims 6, 7, 33-36; Ex.1003, ¶173.

Thus, EDC_525 alone and in combination with EDC_892 and Hanif discloses generating a request to a path calculator (e.g., software instructions for computing backup paths) using the nodes traversed by the end-to-end path of the first backup path for a disjoint second backup path that connects the ingress node to the egress node, which renders obvious this element. Ex.1003, ¶173.

[1.5] *receiving a response from said path calculator; and*

First, as discussed at [1.4], the prior art instructions to compute backup paths, taken separately and together with the processor, corresponds to a “*path calculator*,” as claimed. Ex.1003, ¶174.

Second, EDC_525 render obvious “*receiving a response*.” EDC_525’s instructions for computing determine whether a backup path is available and the level of disjointness, in three different ways. Ex.1003, ¶175.

1. Completely link disjoint example

In the context of FIG. 6, EDC_525 determines the availability of a backup path that is “completely link disjoint” with respect to the other paths from the ingress node to the egress node. Ex.1005, [0035], [0036], FIG. 6; Ex.1005, Claim 33 (“...determine multiple backup paths...[that] are completely link-disjointed with respect to one another.”); *see also* Ex.1005, [0037]-[0039]; Ex.1003, ¶176.

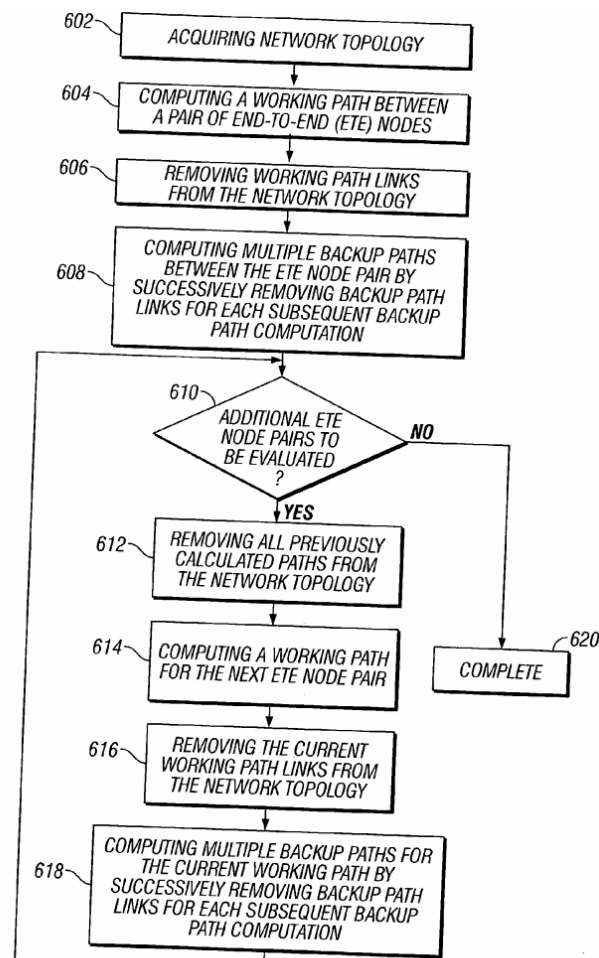
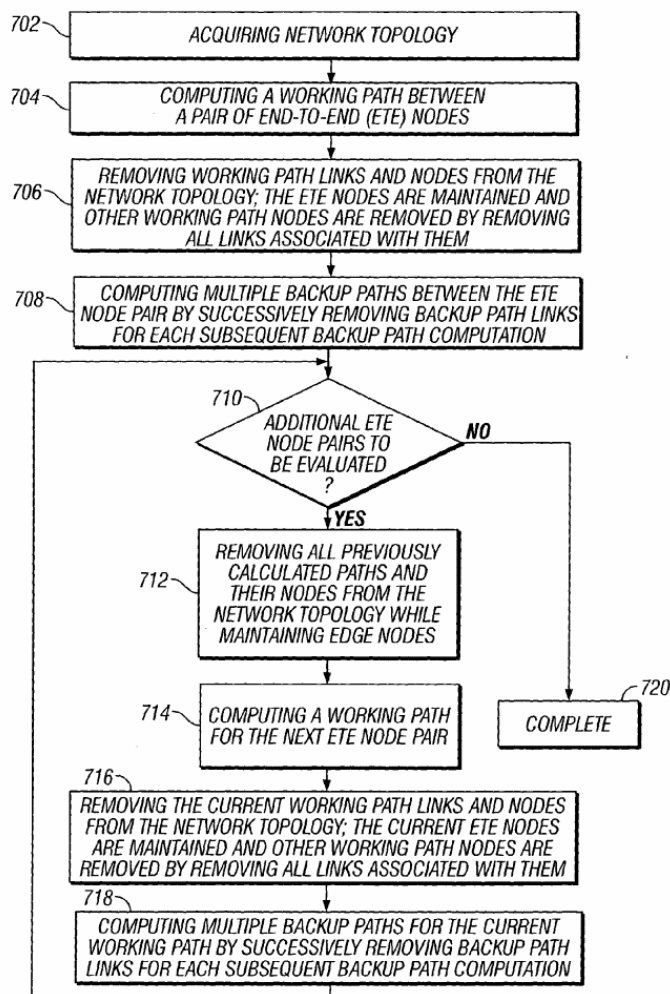


FIG. 6

Ex.1005, FIG. 6.

2. Completely node and link disjoint example

In the context of FIG. 7, EDC_525's path calculator determines the availability of a second backup path “where the links and nodes are completely disjoint” from the ingress node to the egress node. Ex.1005, [0037]-[0038]; *see also* Ex.1005, Claim 21 (“wherein said backup paths comprise paths that are completely node-disjoint with respect to one another.”); Ex.1003, ¶177.



Ex.1005, FIG. 7.

3. Partially disjoint example

In the context of FIG. 8, where it is not feasible to calculate completely disjoint paths, EDC_525's path calculator determines that a “partially disjoint” path from the ingress node to the egress node is available. Ex.1005, [0040]; Ex.1005, Claim 36 (“... said backup paths are partially link-disjointed with respect to one another.”); *see also* Ex.1005, [0041], FIG. 9; Ex.1003, ¶178.

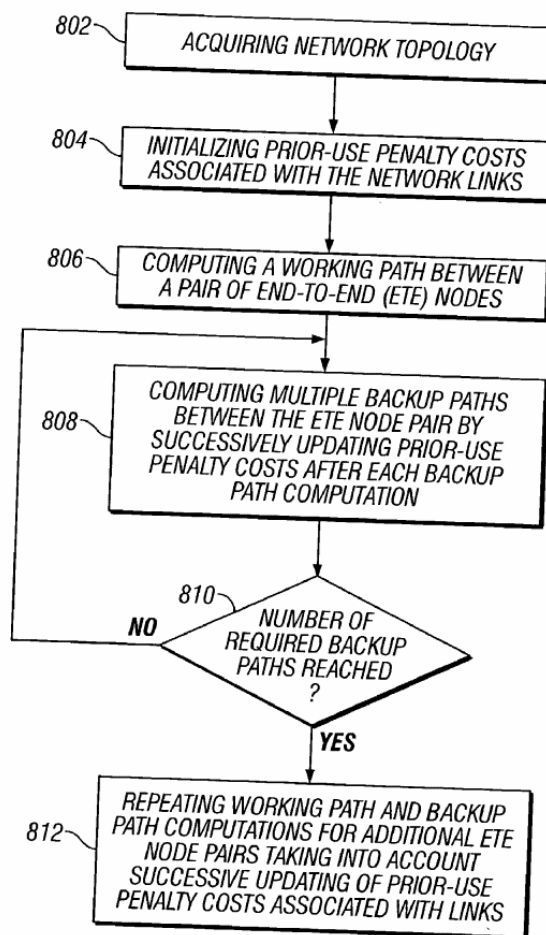


FIG. 8

Ex.1005, FIG. 8.

Consistent with the analysis at [1.4] and knowledge in the art, it would have been obvious to a POSITA for EDC_525's "instructions for computing a plurality of backup paths" (e.g., implemented as a function) to return the result of the path computations (e.g., a determination that a backup path was successful and its level of disjointness). Ex.1003, ¶¶49, 111-112, 170, 179; *see also* Material Facts #13; Ex.1016, 98 ("The **return** keyword exits the function block to the point right after the function call. If return has an argument, that becomes the return value of the function. **You can have more than one return statement in a function.**"); Ex.1033, 380 ("**When the function is called, the operation is performed, and the results are returned.**").

It would have been obvious to a POSITA to receive a response from EDC_525's instructions for computing backup paths that indicates whether the computations were successful and the level of disjointness of the computed backup path (e.g., completely link-disjointed, partially link-disjointed, or both link- and node-disjoint with respect to the first backup path). Receiving an indication that the path computation was successful would beneficially facilitate further action to be taken, including, transmitting appropriate setup/activation messages to the nodes along the path. *See* Ex.1005, [0008], [0031]; Ex.1006, [0036]; *see also, infra*, element [1.6]; Ex.1003, ¶180.

Thus, EDC_525 alone and in combination with EDC_892 and Hanif renders obvious this element. Ex.1003, ¶181.

[1.6] *in response to determining that a fully disjoint path connecting said Point of Local Repair Node to said Merge Point node is available, signaling, to at least one other MPLS label switch router, said fully disjoint path as the Backup LSP to said Bypass LSP.*

First, as discussed at [1.5], it would have been obvious to receive a response that indicated whether the computed backup path is completely link-disjointed, partially link-disjointed, or both link and node disjoint with respect to the established first backup path. Instances where EDC_525's computed second backup path is determined to be available with **“links and nodes [that] are completely disjoint”** (see Ex.1005, [0037]-[0038]) render obvious *“determining that a fully disjoint path connecting said Point of Local Repair Node to said Merge Point node is available.”* Ex.1003, ¶¶182-183.

Second, EDC_525 teaches *“signaling, to at least one other MPLS label switch router,”* by disclosing that **“appropriate setup and/or activation messages may be transmitted from the source node to the path nodes of the network.”** Ex.1005, [0031]; *see also* Ex.1005, [0008]-[0009]; Ex.1006, [0036]. As already analyzed above, EDC_525's GMPLS network nodes correspond to label switch routers. Ex.1005, [0026]-[0027], FIGS. 1, 2. EDC_525's transmission of setup

and/or activation messages to the path nodes renders obvious “*signaling, to at least one other MPLS label switch router,*” as claimed. Ex.1003, ¶184.

Furthermore, EDC_525’s transmission is “*in response to determining...*” because the setup and/or activation messages are transmitted, at least in some instances, “**[o]nce...backup path computations are completed**” and it is determined that a second backup path with “links and nodes [which] are completely disjoint” is available. Ex.1005, [0031], [0037]; *see also* Ex.1005, Claim 21; Ex.1003, ¶185.

Third, EDC_525 teaches signaling the “*fully disjoint path as the Backup LSP to said Bypass LSP.*” EDC_525 provides example information that may be included in transmitted messages and further cites to EDC_892’s “additional details concerning message transmission and wavelength assignment process.” Ex.1005, [0030]-[0031]. In that regard, EDC_892 discloses that the transmitted setup message (also called a “protection message[.]”) identifies that a “**connection is for a protection path**” and also “**identifies the working path that needs the protection.**” Ex.1006, [0033]; *see also* Ex.1006, [0036]; Ex.1003, ¶186.

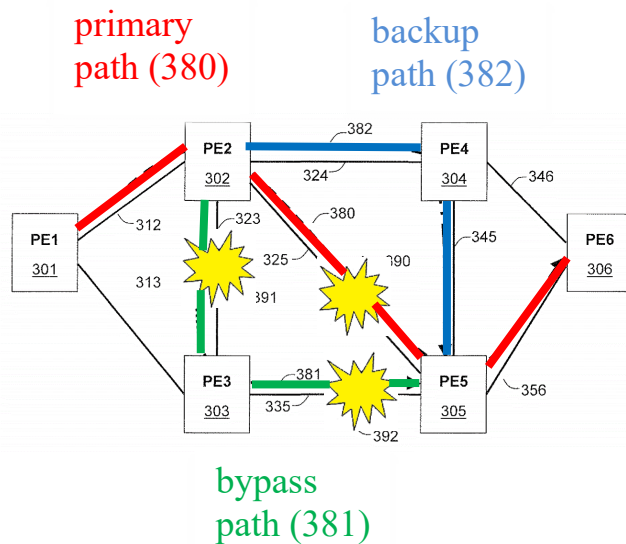
Consistent with the analysis immediately above and at [1.1], [1.4]-[1.5], EDC_525’s second protection path (which may be completely link and node disjoint) provides backup protection in the event that both the working path and the first backup path fail. *See e.g.*, Ex.1005, [0034], Claim 31; Ex.1003, ¶187.

In view of EDC_892's setup message teachings, it would have been obvious for a POSITA to implement EDC_525's setup messages to identify that the completely disjoint second backup path is used to protect the working path that is also protected by the first backup path. Because the second backup path (which is completely disjoint to the first backup path) protects the same working path, the second backup path is the backup to the first backup path when the first backup path fails. Thus, EDC_525 in combination with EDC_892 renders obvious signaling the fully disjoint path "*as the Backup LSP to said Bypass LSP.*"⁵ Ex.1003, ¶188.

As analyzed above, it would have been obvious to combine EDC_892's teachings with EDC_525 because EDC_525 expressly suggests the combination. Ex.1005, [0031]. Additionally, it would have been obvious to a POSITA to include in EDC_525's setup message the noted information, per EDC_892, because it would allow for the nodes that receive the message to update the global allocation database. *See* Ex.1005, [0028], [0030]; Ex.1006, [0033]; *see* Reasons to Combine EDC_892 with EDC_525; Ex.1003, ¶189.

⁵ The prior art's disclosure of both backup paths protecting the same working path is consistent with the claim requirement of "*protecting the Primary LSP against dual failures.*" Ex.1003, ¶188, n2.

As shown in the side-by-side comparison, that EDC_525's second backup path (which is used when the first backup path fails) provides the same protection as the '691 patent's disclosed embodiment. *See also* Ex.1001, 5:48-58; Ex.1005, [0034]; Ex.1003, ¶190.



**Ex.1001, FIG. 3
(annotated).**

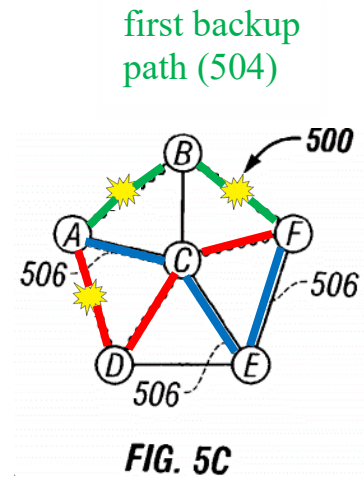


**FIG. 5C
working path (502) second backup path (506)**

**Ex.1005, FIG. 5C
(annotated and modified
to show exemplary failure
condition).**

Thus, EDC_525 in combination with EDC_892 and Hanif discloses transmitting a setup message to the nodes the second backup path (which is completely disjoint) as the backup path in the event that the first backup path fails, which renders obvious this element. Ex.1003, ¶191.

7. Claim 2

[2.0] A method as claimed in claim 1, wherein said path calculator is a constraint based shortest path first calculator.

As discussed at [1.4], the prior art instructions to compute a plurality of backup paths, taken separately and together with the processor, corresponds to “a path calculator.” Further, EDC_525 discloses “a constraint based shortest path first calculator” because the backup paths are calculated using “**resource-based cost constraints**” and “**based on a shortest path first algorithm**.” Ex.1005, Claim 32; *see also* Ex.1005, [0008], [0032]-[0033]. Additionally, EDC_525’s calculation is also constrained because previously computed paths (e.g., nodes and/or links) are “logically removed” from the topology for calculation purposes or penalized by a “variable cost factor.” *See* Ex.1005, [0032], [0040]. Ex.1003, ¶¶192-193.

Additionally, EDC_892 discloses calculating a “**constrained shared protection path...[that is] the shortest path**.” Ex.1006, [0047]; *see also* Ex.1006, [0038], [0046]-[0047], Abstract. It would have been obvious to a POSITA to implement EDC_525’s path calculations as disclosed by EDC_892, in view of EDC_525’s express citation to the exemplary path calculations of EDC_892. Ex.1005, [0031]; Ex.1003, ¶¶194-196.

8. Claim 3

[3.0] A method as claimed in claim 1, wherein in response to determining that the fully disjoint path connecting said Point of Local Repair Node to said Merge Point node is not available, in response to determining that a partially disjoint path connecting said Point of Local Repair Node to said Merge Point node is available, signaling, to at least one other MPLS label switch router, said partially disjoint path as the Backup LSP to said Bypass LSP.

EDC_525 discloses that “[b]ecause of resource constraints and connectivity blocking in a network, it may not be feasible to treat the links, nodes, or both, in a completely disjointed fashion for calculating the backup paths.” Ex.1005, [0040]. EDC_525’s disclosure that it may not be feasible to treat links and nodes as completely disjoint renders obvious “*determining that the fully disjoint path connecting said Point of Local Repair Node to said Merge Point node is not available.*” Ex.1003, ¶197.

EDC_525 discloses that in instances where completely disjoint paths are not feasible, a “variable cost factor may be employed to penalize the links and/or nodes already used for a connection...[when] computing multiple backup paths where the links are partially disjoint.” Ex.1005, [0040], FIG. 8; *see also* 1005, [0041]. Further, as discussed at [1.4]-[1.5], the instructions for computing backup paths may be dynamically invoked and based on the result (which in this case would identify backup paths that are partially disjoint), appropriate setup and/or

activation messages may be transmitted to the nodes along the path as discussed at [1.6]. Ex.1003, ¶198.

9. Claim 4

[4.0] A method as claimed in claim 3, wherein in response to determining that a partially disjoint path connecting said Point of Local Repair Node to said Merge Point node is not available, then signaling an error on the attempt to provide a Backup LSP.

Hanif discloses that if “**a local repair path could not be found....An error condition may then be output** indicating that a local repair LSP could not be established.” Ex.1013, [0050]. A POSITA would have recognized that Hanif’s disclosed situation (where a local repair path cannot be found) is an example within the scope of claim 4’s conditional language. If no local repair path is available (irrespective of disjointedness), then it necessarily follows that “*a partially disjoint path... is not available.*” It would have been obvious to a POSITA to apply Hanif’s teachings to EDC_525 and output an error if it is determined a second backup path connecting the ingress node and the egress node is unavailable, to notify a network administrator that corrective action should be taken. *See e.g.*, Ex.1020, 8:50-54; *see also* Reasons to Combine Hanif with EDC_525. Ex.1003, ¶¶199-200.

10. Claim 5

[5.0] A method as claimed in claim 1 after said obtaining step, comprising further steps of: procuring a Shared Risk Link Groups (SRLG) associated with the nodes traversed by the end-to-end path of said Bypass LSP from said Point of Local Repair Node to said Merge Point node; and providing said Shared Risk Link Groups as part of said generating a request step to said calculator for use in calculating said disjoint path.

Consistent with [1.0.2], EDC_525 provides a further embodiment that monitors the network to define a SRLG of the working and first backup paths. Ex.1005, [0042]. An SRLG defined in that way would include the nodes traversed by the end-to-end path of the first backup path from the ingress node to the egress node (as discussed at [1.2]). Accordingly, EDC_525 renders obvious “*procuring a Shared Risk Link Groups (SRLG) associated with the nodes traversed by the end-to-end path of said Bypass LSP from said Point of Local Repair Node to said Merge Point node.*” Ex.1003, ¶¶201-202.

Furthermore, EDC_525 discloses that paths that belong to the defined SRLG are “not used” for the backup path and that appropriate computation schemes (e.g., the complete link disjoint methodology) should be used “to reduce the possibility of multiple failures.” Ex.1005, [0042]. It would have been obvious for a POSITA to provide the defined SRLG as part of the generated path-computation request (See [1.4]) because this information facilitates computing backup paths whose links do not belong to the same SRLG (e.g., a “complete link disjoint

methodology”). This, in turn, reduces the possibility of simultaneous working-path and backup-path failures. Thus, EDC_525 in combination with EDC_892 and Hanif renders obvious “*providing said Shared Risk Link Groups as part of said generating a request step... for use in calculating said disjoint path.*” Ex.1003, ¶¶203-204.

11. Claim 6

Claims 6-10 merely repeat the subject matter of claims 1-5 but are phrased as computer-readable-medium claims instead of method claims. Accordingly, claims 6-10 would have been obvious for the same reasons as claims 1-5 discussed above. Nevertheless, analysis is provided below regarding the minor wording differences between the claim sets.

[6.0.1] A non-transitory machine readable storage medium encoded with instructions for execution by a network processor of a Multiprotocol Label Switching (MPLS) label switch

To the extent limiting, consistent with [1.0.1], EDC_525 alone and in combination with EDC_892 and Hanif discloses a processor of a “switching” node in an MPLS network, which renders obvious “*a network processor of a Multiprotocol Label Switching (MPLS) label switch.*” Ex.1005, [0025].

Additionally, Hanif discloses that the network node “**may be embodied as a network device such as a switch or router.**” Ex.1013, [0066]. It would have been obvious to a POSITA to implement EDC_525’s network node as a switch, per

Hanif, because this is merely a simple substitution of one known element for another (EDC_525's node for Hanif's switch) to obtain predictable results (perform switching in the network). The combination is also merely a combination of prior art elements (Hanif's switch with EDC_525's network node) according to known methods (it was known how to implement switches) to yield predictable results (perform switching). *See also* Reasons to combine Hanif with EDC_525. Ex.1003, ¶205.

The prior art combination also renders obvious “*non-transitory machine readable storage medium encoded with instructions for execution.*” EDC_525 further discloses “**processor-accessible medium having a plurality of instructions for carrying out network operations.**” Ex.1005, Claim 31; *see also* Ex.1005, [0009]; Ex.1003, ¶206. Consistent with knowledge in the art, it would have been obvious to a POSITA for EDC_525's processor-accessible medium to be implemented as “*non-transitory*” medium (e.g., flash memory, system memory, hard drive, etc.), because such an implementation would allow for retaining the instructions after a power down. Ex.1011, 6:24-30, Claim 12. Consistent with [1.0.1], it would have been obvious to a POSITA for EDC_525's instructions to be for “*execution*” by a processor, as was conventional and well known in the art. Ex.1012, Claim 17; Ex.1003, ¶206.

Hanif discloses that the memory includes “**instructions that are executed by processor.**” *See e.g.*, Ex.1013, [0074]; *see also* Ex.1013, [0060], [0073]. It would have been obvious to a POSITA to apply Hanif’s teachings and implement EDC_525’s instructions so that they are executable by a processor to facilitate the computations of multiple backup paths and other network operations, thereby furthering EDC_525’s objectives. *See e.g.*, Ex.1005, [0034]; *see also* Reasons to Combine Hanif with EDC_525; Ex.1003, ¶¶207-208.

Element [6.0.2]

See [1.0.2], [6.0.1]; Ex.1003, ¶209.

Element [6.1]

See [1.1], [6.0.1]-[6.0.2]; Ex.1003, ¶210.

Element [6.2]

See [1.2], [6.0.1]-[6.0.2]; Ex.1003, ¶211.

Element [6.3]

See [1.3], [6.0.1]-[6.0.2]; Ex.1003, ¶212.

Element [6.4]

See [1.4], [6.0.1]-[6.0.2]; Ex.1003, ¶213.

Element [6.5]

See [1.5], [6.0.1]-[6.0.2]; Ex.1003, ¶214.

Element [6.6]

See [1.6], [6.0.1]-[6.0.2]; Ex.1003, ¶215.

12. Claim 7

See [6.0.1]-[6.6], [2.0]; Ex.1003, ¶216.

13. Claim 8

See [6.0.1]-[6.6], [3.0]; Ex.1003, ¶¶217-218.

14. Claim 9

See [6.0.1]-[6.6], [4.0]; Ex.1003, ¶219.

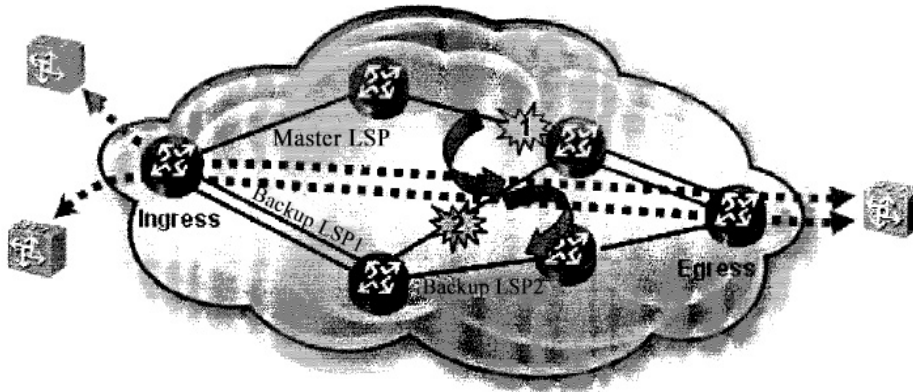
15. Claim 10

See [6.0.1]-[6.6], [5.0] Ex.1003, ¶220-221.

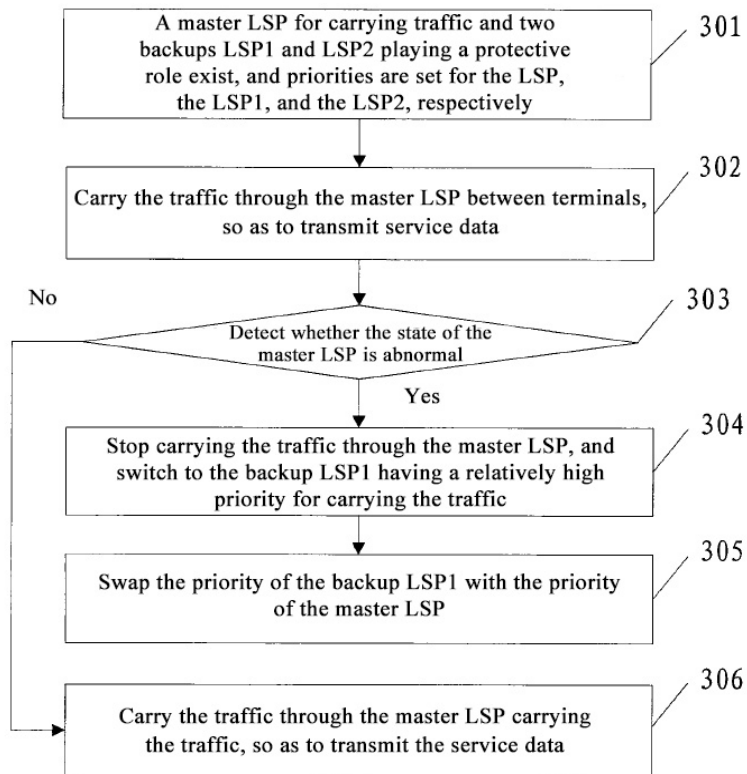
D. Ground 2: Claims 1-10 are Obvious over EDC_525, EDC_892, Hanif, and Li.

1. Li

Li describes prioritizing backup LSPs in MPLS networks such that a higher-priority backup LSP is initially utilized when a primary (or master) LSP has an abnormality (e.g., link failure, etc.) and a lower-priority LSP is utilized in case the higher-priority backup LSP fails. Ex.1030, 6-7, Abstract, FIGS. 3, 4; Ex.1003, ¶¶222-224; *see also* Ex.1030, 8-9, FIG. 5-6; Ex.1003, ¶¶225-226.



Ex.1030, FIG. 3.



Ex.1030, FIG. 4.

2. Reasons to Combine Li with EDC_525

a. Li is Analogous Art

Li is analogous art to the '691 patent because it pertains to the same field of

MPLS communications. Ex.1003, ¶227; Ex.1001, 1:6-10, Abstract; Ex.1030, Abstract. Li is also analogous art because it is reasonably pertinent to the problem of using multiple backup paths to protect against multiple failures. *See* Ex.1001, 1:6-10, Abstract; Ex.1030, 6-7, Abstract, FIGS. 3-4; Ex.1003, ¶227.

b. Motivation to combine with Li

A POSITA would have been motivated to combine the teachings of Li and EDC_525 (as modified in view of EDC_892 and Hanif) to produce numerous predictable and beneficial results. Ex.1003, ¶228.

EDC_525 teaches that two backup paths provide protection for each other in case of multiple simultaneous failures. *See e.g.*, Ex.1005, [0034]. EDC_525, however, leaves it to a POSITA to determine the sequence of backup path utilization. As such, a POSITA looking to implement EDC_525's teachings would have looked to other relevant teachings for this purpose. Li discloses prioritizing two backup paths and using the higher-priority backup path first, then using the lower-priority backup path in case that path also fails. *See* Ex.1030, 6. For example, in the context of FIGS. 3 and 4, Li discloses that a higher-priority backup path (backup LSP1) is initially utilized when a primary path (master LSP) fails and, in the event that the higher-priority path subsequently fails, then a lower-priority backup path (backup LSP2) is utilized. Ex.1030, 6-7, FIGS. 3 and 4; *see also* Ex.1030, 9-10, FIGS. 5-6; Ex.1003, ¶229.

It would have been obvious to a POSITA to apply Li's teachings when implementing multiple backup paths, per EDC_525, such that failover sequence would be controlled based on priority (i.e., with a higher-priority backup path being used first in case of failure and a lower-priority backup path being used if the higher-priority path also fails). Such an implementation would beneficially give greater control over path failover selection. Ex.1003, ¶230.

Also, a POSITA would have been motivated to apply Li's priority teachings, when implementing two backup paths per EDC_525, because it may reduce the probability that the selected backup path is preempted. For example, it was known in the art to include priority fields when a path is set up for FRR (or local protection) such that a path with a higher priority has less likelihood that it would be preempted by other paths, as compared with a backup path having a lower priority. Ex.1019, 8-9; Ex. 1025, [0043]-[0044]; Ex.1024, [0055]. Accordingly, it would have been obvious to a POSITA implementing two backup paths, per EDC_525, to assign priorities to backup paths and upon a failure of a working path use a backup path with the higher priority (as Li teaches) because this would reduce the probability that the backup path is preempted by another path during operation. If a backup path with a lower priority were used initially, the probability of preemption would be greater, which would have been recognized as undesirable. Ex.1003, ¶231.

Additionally, a POSITA would have been motivated to utilize Li's priority scheme for backup path selection, when implementing two backup paths per EDC_525, because it would allow for assigning a first backup path a high priority when it has the same (or greater) quality of service ("QoS") as the working path, thereby ensuring that the QoS is maintained even though failure has occurred. Ex.1003, ¶232. Maintaining QoS would have been recognized as important by a POSITA for paths that carry voice data, among other sensitive data. *See* Ex.1021, 1:22-26; Ex.1026, 7; Ex.1027, 6; Ex.1030, 1; Ex.1003, ¶232.

Yet another reason to combine the teachings of Li with EDC_525 is that prioritization allows for using the priority levels of the different paths to minimize frequent switching when the primary path is restored to a normal state. *See* Ex.1030, 1. Path prioritization "solve[s] the problem of severe impact on traffic reliability caused by frequent traffic switching when a network is unstable" because the traffic can be "locked on and carried through the backup path that is upgraded to the master path, without being switched back to the original master path." Ex.1030, 1-2. Ex.1003, ¶233.

Each above noted reason, separately and together would have motivated a POSITA to apply Li's priority teachings to EDC_525. Ex.1003, ¶234.

Furthermore, as discussed in Ground 1 element [1.0.2] and [1.2], EDC_525 teaches that the path computations may be performed at a central node. Ex.1005,

[0025] (“...centrally...”); *see also* Ex.1005, [0042] (“...centralized...”). It would have been obvious to a POSITA, in implementing the combination of Li with EDC_525 where the path computations are performed centrally to include this information in a signal sent by the central node to the ingress node (point of local repair). *See* Ex.1025, [0043]-[0044] (“...a FAST_REROUTE object used for setup of a backup LSP for fast rerouting [includes].... A Setup Priority field 200 contains a value representing priority of a backup LSP...A Holding Priority field 202...”); Ex.1019, 8-9 (“...The FAST_REROUTE object is used to control the backup used for the protected LSP. This specifies the setup and hold priorities.”); Ex.1024, [0055] (“The Path message carries...establishment priority, hold priority...”). The signal would inform the ingress node of the calculated backup paths that need to be setup and the sequence of backup path failover. Ex.1003, ¶235.

The combination of Li with EDC_525 represents a simple combination of known elements (e.g., path priority per Li with EDC_525’s paths) to yield predictable results (e.g., informing an ingress node which backup path to use first if a working path fails). The proposed combination also merely represents the application of a known technique (e.g., assigning path priority, per Li, to EDC_525’s paths) to yield predictable and beneficial results (e.g., informing the ingress node which backup path to use first, and obtaining one or more of the noted benefits discussed immediately above). Ex.1003, ¶236.

c. Expectation of success

The results would have been predictable and there would have been a reasonable expectation of success in the combination since EDC_525 and Li address the same technology, as analyzed above. Also, path prioritization was well-known in the art, and it was known how to provide path priority information in signals, which further supports reasonable expectation of success. Material Facts #14-15; Ex.1003, ¶237.

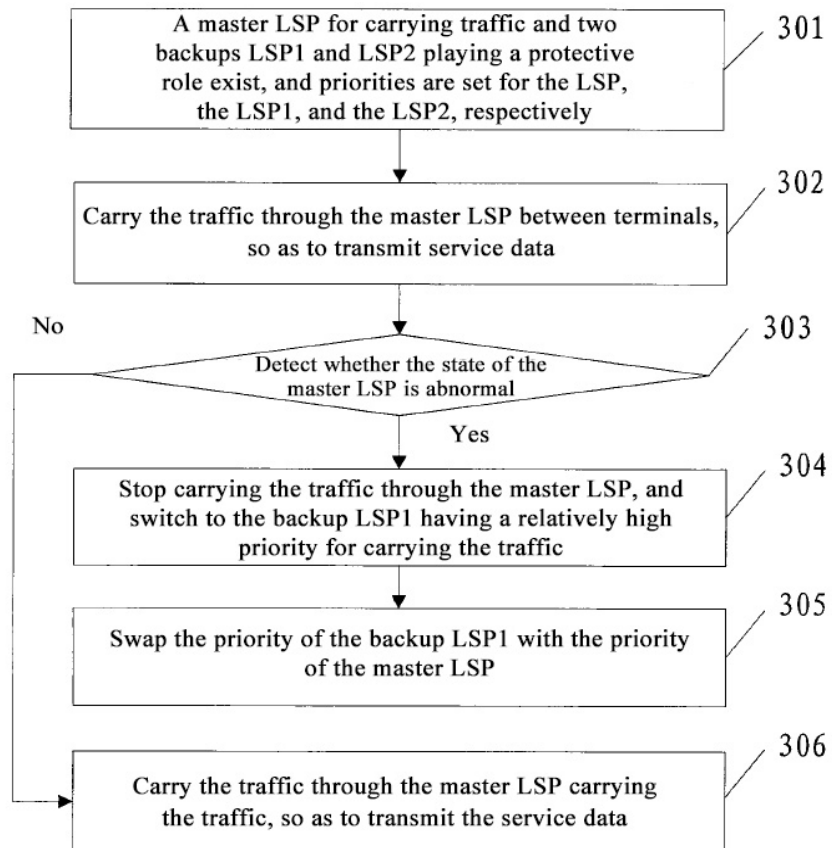
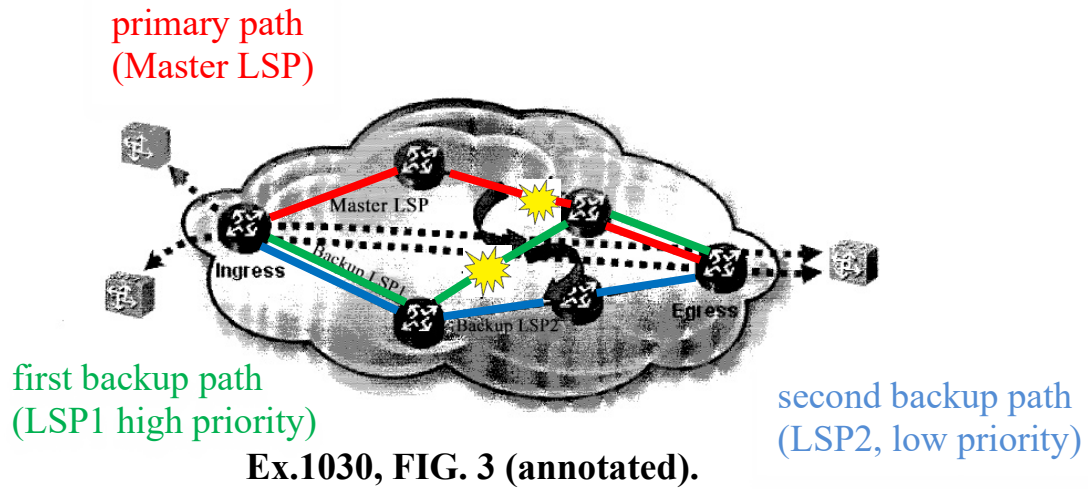
3. Claim 1

The Ground 2 analysis of claim 1 is the same as in Ground 1 except as identified below. Ex.1003, ¶238.

To the extent argued that “*signaling... said fully disjoint path as the Backup LSP to said Bypass LSP*” is not rendered obvious per the Ground 1 analysis, the further combination with Li renders it obvious. Ex.1003, ¶239.

EDC_525 teaches that two backup paths provide protection for each other in case of multiple simultaneous failures. Ex.1005, [0034]. Ex.1003, ¶241. Li discloses prioritizing two backup paths and using the higher-priority backup path first when there is a failure. Ex.1030, 6. In the context of FIGS. 3 and 4, Li discloses that a higher-priority backup path (backup LSP1) is initially utilized when a primary path (master LSP) fails and, in the event that the higher-priority path subsequently fails, then a lower-priority backup path (backup LSP2) is

utilized. Ex.1030, 6-7, FIGS. 3 and 4; *see also* Ex.1030, 9-10, FIGS. 5-6; Ex.1003,
¶242.



It would have been obvious to a POSITA to apply Li's teachings when implementing EDC_525's multiple backup paths to provide control over the failover sequence (e.g., with a higher-priority backup path being preferred after a working-path failure, and a lower-priority backup being used if the higher-priority backup path has also failed). Ex.1003, ¶243. Such an implementation would beneficially give greater control over path failover. Also, a POSITA would have been motivated to utilize Li's priority scheme teachings, when implementing two backup paths per EDC_525, because it would provide one or more of the noted benefits discussed above. *See* Reasons to Combine Li and EDC_525; Ex.1003, ¶243.

Furthermore, consistent with the analysis of [1.0.2] and [1.2] in Ground 1, EDC_525 teaches that the path computations may be performed at a central node. Ex.1005, [0025] ("...centrally..."); *see also* Ex.1005, [0042] ("...centralized..."). It would have been obvious to a POSITA implementing the combination of Li with EDC_525, when the path computations for the second backup path are performed centrally, for the central node to send a signal (e.g., setup signal, see Ground 1 element [1.6]) to the ingress node (point of local repair) indicating that a fully disjoint second backup path needs to be setup. Ex.1003, ¶244. It also would have been obvious to a POSITA, consistent with knowledge in the art, to include priority information to inform the ingress node about the backup path's priority

and the sequence to use the backup paths in case the working path fails. *See, e.g.*, Ex. 1025, [0043]-[0044]; Ex.1019, 8-9; Ex.1024, [0055]; Ex.1003, ¶244.

For example, the second backup path may have a lower QoS and therefore be given a lower priority than the (already established) first backup path. If the working path fails, the first backup path would initially provide protection at the same QoS as the working path. The second backup path would be used only in the circumstance where the first backup path also fails. *See* Ex.1021, 1:22-26; Ex.1026, 7; Ex.1027, 6; Ex.1003, ¶245.

Thus, EDC_525 in combination with EDC_892, Hanif, and Li discloses transmitting a setup message from a central node to the ingress node that identifies that the fully disjoint second backup path has a lower priority than the first backup path, which renders obvious this element. Ex.1003, ¶246.

4. Claims 2-5

See Claim 1 in Ground 2 and Claims 2-5 in Ground 1. Ex.1003, ¶247.

5. Claim 6

Other than the below identified element, the remaining Claim 6 analysis in Ground 1 remains the same in Ground 2. Ex.1003, ¶248.

EDC_525 in combination with EDC_892, Hanif, and Li renders obvious “*signaling, to at least one other MPLS label switch router, said fully disjoint path*

as the Backup LSP to said Bypass LSP” for the reasons discussed above at Claim 1 in Ground 2. Ex.1003, ¶249.

6. Claims 7-10

See Claim 6 in Ground 2 and Claims 7-10 in Ground 1. Ex.1003, ¶250.

X. CONCLUSION

Petitioner has established a reasonable likelihood that the Challenged Claims are unpatentable.

Respectfully submitted,

Dated: November 15, 2024
HAYNES AND BOONE, LLP
2801 N. Harwood St., Suite 2300
Dallas, Texas 75201
Customer No. 27683

/Theodore M. Foster/
Theodore M. Foster
Lead Counsel for Petitioner
Registration No. 57,456

XI. MANDATORY NOTICES

A. Real Party-in-Interest

Pursuant to 37 C.F.R. § 42.8(b)(1), Petitioner certifies that the real party-in-interest is Cisco Systems, Inc.

B. Related Matters

Pursuant to 37 C.F.R. § 42.8(b)(2), to the best knowledge of the Petitioner, the '691 patent is involved in the following cases:

Case Heading	Number	Court	Date
<i>WSOU Investments, LLC v. Cisco Systems, Inc.</i>	2-24-cv-00332	EDTX	May 6, 2024

C. Lead and Back-up Counsel and Service Information

Lead Counsel

Theodore M. Foster
HAYNES AND BOONE, LLP
2801 N. Harwood St., Suite 2300
Dallas, TX 75201

Phone: (303) 382-6205
Fax: (214) 200-0853
ipr.theo.foster@haynesboone.com
USPTO Reg. No. 57,456

Back-up Counsel

David L. McCombs
HAYNES AND BOONE, LLP
2801 N. Harwood St., Suite 2300
Dallas, TX 75201

Phone: (214) 651-5533
Fax: (214) 200-0853
david.mccombs.ipr@haynesboone.com
USPTO Reg. No. 32,271

Gregory P. Huh
HAYNES AND BOONE, LLP
2801 N. Harwood St., Suite 2300
Dallas, TX 75201

Phone: (972) 739-6939
Fax: (214) 200-0853
gregory.huh.ipr@haynesboone.com
USPTO Reg. No. 70,480

Please address all correspondence to lead and back-up counsel. Petitioner consents to service in this proceeding by email at the addresses above.

XII. CLAIMS APPENDIX

- [1.0.1] 1. A method performed by a network processor of a Multiprotocol Label Switching (MPLS) label switch router
- [1.0.2] for providing a Backup Label Switched Path (LSP) to a Bypass LSP already established for a Protected Primary LSP, the method comprising the steps of:
- [1.1] protecting the Primary LSP against dual failures, comprising:
- [1.2] establishing the Bypass LSP for the Protected Primary LSP having a Point of Local Repair node and a Merge Point node;
- [1.3] obtaining the nodes traversed by an end-to-end path of said Bypass LSP from said Point of Local Repair Node to said Merge Point node;
- [1.4] generating a request to a path calculator using the nodes traversed by said end-to-end path of said Bypass LSP for a disjoint path connecting said Point of Local Repair Node to said Merge Point node;
- [1.5] receiving a response from said path calculator; and
- [1.6] in response to determining that a fully disjoint path connecting said Point of Local Repair Node to said Merge Point node is available, signaling, to at least one other MPLS label switch router, said fully disjoint path as the Backup LSP to said Bypass LSP.
- [2.0] 2. A method as claimed in claim 1, wherein said path calculator is a constraint based shortest path first calculator.

- [3.0] 3. A method as claimed in claim 1, wherein in response to determining that the fully disjoint path connecting said Point of Local Repair Node to said Merge Point node is not available, in response to determining that a partially disjoint path connecting said Point of Local Repair Node to said Merge Point node is available, signaling, to at least one other MPLS label switch router, said partially disjoint path as the Backup LSP to said Bypass LSP.
- [4.0] 4. A method as claimed in claim 3, wherein in response to determining that a partially disjoint path connecting said Point of Local Repair Node to said Merge Point node is not available, then signaling an error on the attempt to provide a Backup LSP.
- [5.0] 5. A method as claimed in claim 1 after said obtaining step, comprising further steps of: procuring a Shared Risk Link Groups (SRLG) associated with the nodes traversed by the end-to-end path of said Bypass LSP from said Point of Local Repair Node to said Merge Point node; and providing said Shared Risk Link Groups as part of said generating a request step to said calculator for use in calculating said disjoint path.
- [6.0.1] 6. A non-transitory machine readable storage medium encoded with instructions for execution by a network processor of a Multiprotocol Label Switching (MPLS) label switch

[6.0.2] for providing a Backup Label Switched Path (LSP) to a Bypass LSP already established for a Protected Primary LSP, the medium comprising:

[6.1] instructions for protecting the Primary LSP against dual failures, comprising:

[6.2] instructions for establishing the Bypass LSP for the Protected Primary LSP having a Point of Local Repair node and a Merge Point node;

[6.3] instructions for obtaining the nodes traversed by an end-to-end path of said Bypass LSP from said Point of Local Repair Node to said Merge Point node;

[6.4] instructions for generating a request to a path calculator using the nodes traversed by said end-to-end path of said Bypass LSP for a disjoint path connecting said Point of Local Repair Node to said Merge Point node;

[6.5] instructions for receiving a response from said path calculator; and;

[6.6] in response to determining that a fully disjoint path connecting said Point of Local Repair Node to said Merge Point node is available, instructions for signaling, to at least one other MPLS label switch router, said fully disjoint path as the Backup LSP to said Bypass LSP.

[7.0] 7. A non-transitory machine readable storage medium as claimed in claim 6, wherein the instructions specify that said path calculator is a constraint based shortest path first calculator.

[8.0] 8. A non-transitory machine readable storage medium as claimed in claim 6, wherein in response to determining that the fully disjoint path connecting

said Point of Local Repair Node to said Merge Point node is not available, in response to determining that a partially disjoint path connecting said Point of Local Repair Node to said Merge Point node is available, signaling, to at least one other MPLS label switch router, said partially disjoint path as the Backup LSP to said Bypass LSP.

[9.0] 9. A non-transitory machine readable storage medium as claimed in claim 6, wherein in response to determining that a partially disjoint path connecting said Point of Local Repair Node to said Merge Point node is not available, signaling an error on the attempt to provide a Backup LSP.

[10.0] 10. A non-transitory machine readable storage medium as claimed in claim 6, wherein the instructions specify after said obtaining step further steps of: procuring a Shared Risk Link Groups (SRLG) associated with the nodes traversed by the end-to-end path of said Bypass LSP from said Point of Local Repair Node to said Merge Point node; and providing said Shared Risk Link Groups as part of said generating a request step to said calculator for use in calculating said disjoint path.

CERTIFICATE OF WORD COUNT

Pursuant to 37 C.F.R. § 42.24(d), Petitioner hereby certifies, in accordance with and in reliance on the word count provided by the word-processing system used to prepare this Petition, that the number of words in this paper is 13,782.

Pursuant to 37 C.F.R. § 42.24(d), this word count excludes the table of contents, table of authorities, mandatory notices under § 42.8, certificate of service, certificate of word count, appendix of exhibits, and any claim listing.

Dated: November 15, 2024

/Theodore M. Foster/
Theodore M. Foster
Lead Counsel for Petitioner
Registration No. 57,456

CERTIFICATE OF SERVICE

The undersigned certifies that, in accordance with 37 C.F.R. § 42.6(e) and 37 C.F.R. § 42.105, service was made on Patent Owner as detailed below.

Date of service November 15, 2024

Manner of service FEDERAL EXPRESS

Documents served Petition for *Inter Partes* Review Under 35 U.S.C. § 312 and 37 C.F.R. § 42.104 of U.S. 8,982,691; Petitioner's Exhibit List; All Exhibits; Petitioner's Power of Attorney.

Persons served Terry W. Kramer, Esq.
Kramer & Amado, P.C.
225 Reinekers Ln., Suite 300
Alexandria, VA 22314

/Theodore M. Foster/
Theodore M. Foster
Lead Counsel for Petitioner
Registration No. 57,456