

**AREE WITIN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
SHERMAN DIVISION**

**INTELLECTUAL VENTURES I LLC AND
INTELLECTUAL VENTURES II LLC,**

Plaintiff,

vs.

AMERICAN AIRLINES, INC.,

Defendant.

Civil Action No. 4:24-cv-00980

JURY TRIAL

DEFENDANT AMERICAN AIRLINES, INC.’S P.R. 3-3 INVALIDITY CONTENTIONS

Pursuant to P.R. 3-3, P.R. 3-4, and the Court’s Scheduling Order in the above-captioned case (Dkt. 44, 49), Defendant American Airlines, Inc. (“American”) serves its Preliminary Invalidity Contentions with respect to the claims identified by Plaintiff Intellectual Ventures I LLC and Intellectual Ventures II LLC (collectively “IV”) in its Disclosure of Asserted Claims and Infringement Contentions, served May 23, 2025 (“Infringement Contentions”).

These Preliminary Invalidity Contentions are limited to the claims asserted from the six patents identified in IV’s Original Complaint (Dkt. 1). The Court has not yet ruled on IV’s Opposed Motion for Leave to File an Amended Complaint. *See* Dkt. 48. If the Court grants IV’s Motion, American reserves the right to supplement these contentions to raise invalidity and/or unenforceability contentions directed to any new or additional asserted claims.

I. PRELIMINARY STATEMENT AND RESERVATION OF RIGHTS

A. The Asserted Claims

IV asserts the following patents, claims, and priority dates in its Infringement Contentions, served May 23, 2025.

Asserted Patent	Asserted Claims	Asserted Priority Date(s)
'844 patent	7 and 11	December 30, 2004
'722 patent	14, 16, 17, and 18	December 18, 2000
'785 patent	30, 35, and 37	March 31, 2003
'326 patent	1, 4, and 18	January 12, 2004
'469 patent	24, 25, 26, 28, and 32	September 29, 2003
'582 patent	1, 2, 5, and 7	March 13, 2002

The patents identified in the foregoing table are collectively referred to as the “Asserted Patents,” and the claims identified in the foregoing table are collectively referred to as the “Asserted Claims.” Any reference to an “asserted priority date” in these Preliminary Invalidation Contentions refers to the “Asserted Priority Dates” identified in the foregoing table, and further defined below.

American contends that each of the Asserted Claims is invalid under at least one or more of 35 U.S.C. §§ 101, 102, 103, or 112. Pursuant to the Local Patent Rules, American does not provide any contentions regarding any claims not asserted by IV. To the extent that the Court permits IV to assert additional claims (or patents) against American in the future, American reserves all rights to amend or supplement these Preliminary Invalidation Contentions or to otherwise disclose new or supplemental invalidity contentions regarding such claims. Furthermore, because discovery is ongoing,¹ American reserves the right to revise, amend,

¹ Defendant’s ongoing efforts include but are not limited to: serving subpoenas on prior artists and inventors regarding prior art and claim scope, seeking additional information related to the references and systems disclosed in these Preliminary Invalidation Contentions, and seeking additional information related to available prior art systems, as well as IV’s Infringement

and/or supplement the information provided herein, including identifying, charting, and relying on additional references, should discovery yield additional information or references.²

American further reserves the right to amend these contentions in response to any claim construction rulings, as permitted by the Local Patent Rules or with permission of the Court.

The Infringement Contentions are deficient in multiple respects and do not provide American with sufficient information to understand the specific accused features and components and the alleged factual and evidentiary bases for IV's infringement allegations. Among other things, the Infringement Contentions lack the specificity required by P.R. 3-1, fail to properly identify accused instrumentalities, and fail to explain adequately IV's infringement theories for numerous limitations. IV has prejudiced American's ability to understand, for purposes of preparing these Preliminary Invalidity Contentions, what IV alleges to be the scope of the Asserted Claims. If IV modifies any assertion or contention in its Infringement Contentions, or presents any new assertion or contention relevant to these Preliminary Invalidity Contentions to the extent allowed by the Local Patent Rules or the Court, American reserves the right to supplement or otherwise amend these initial Invalidity Contentions.

B. Priority Date of the Patents-in-Suit

IV's Infringement Contentions contain allegations regarding the priority date to which IV alleges it is entitled for each of the Asserted Claims. American does not agree that IV is entitled

Contentions and the products accused of infringing therein. No depositions have been taken as of this time, including, without limitation, depositions of any inventors, authors, or entities listed on any references or systems identified in these Preliminary Invalidity Contentions. Further, Defendant reserves the right to review and supplement these Preliminary Invalidity Contentions with respect to any additional prior art that becomes apparent as discovery proceeds.

² IV has failed to respond to American's Interrogatory No. 5, seeking the identification of all allegations by any Person or Entity that any claim of the Patents-in-Suit or any Related Patent is invalid or unenforceable, and Interrogatory No. 6, seeking all prior art of which IV is aware or that anyone has ever contended or asserted to be prior art to the Asserted Claims. American incorporates by reference any responses to those interrogatories IV may provide in the future.

to the Asserted Priority Dates for each of the Asserted Claims, as IV has failed to prove it is entitled to its Asserted Priority Dates. Furthermore, IV has failed to meaningfully respond to American's Interrogatory No. 7, seeking details regarding the conception and reduction to practice of the Asserted Claims.

Any reference to an "asserted priority date" in these Preliminary Invalidity Contentions refers to the priority dates identified in IV's Infringement Contentions. Reference to a "priority date" or an "asserted priority date" should not be construed to mean that American agrees that any of the Asserted Patents are in fact entitled to such priority date, or that IV has provided proper notice as to its contentions for a priority date.³ To the extent IV alleges that any prior art relied on in these Preliminary Invalidity Contentions does not actually qualify as prior art to an Asserted Patent, American reserves the right to rebut those allegations (e.g., by demonstrating an earlier priority date for the challenged prior art and/or a later priority date for a particular Asserted Patent and/or Asserted Claim).⁴ Likewise, to the extent IV successfully establishes an invention date before any of the prior-art references relied on by American, then those references serve as evidence of obviousness, particularly, contemporaneous invention by others.

C. Claim Construction

American's Preliminary Invalidity Contentions are based on (1) American's present understanding of the Asserted Claims, (2) the claim constructions IV appears to be proposing based on the Infringement Contentions, all without regard to whether American agrees with IV's apparent or expressed claim constructions. American reserves the right to supplement or otherwise amend these Preliminary Invalidity Contentions in response to any proposed claim

³ American reserves the right to rely on additional documents and evidence, including without limitation the documents cited in IV's Infringement Contentions, in the event that IV fails to establish that any Asserted Claim of any Asserted Patent is entitled to its Asserted Priority Date.

⁴ American reserves the right to rely on additional documents and evidence to rebut any efforts by IV to allege any reference was not publicly available or otherwise available as prior art.

constructions or alleged supporting evidence offered by IV, any report from any expert witness for IV regarding claim construction issues, any claim construction briefing filed by IV, and any position taken by IV concerning claim construction, infringement, or invalidity.

American takes no position on any matter of claim construction in these Preliminary Invalidation Contentions. If American's apparent claim constructions herein are consistent with any explicit, apparent, or implied claim constructions in the Infringement Contentions, no inference is intended and no inference should be drawn that American agrees with any of IV's claim constructions. Any statement herein describing or tending to describe any claim element is provided solely for the purpose of understanding and/or applying the cited prior art. American expressly reserves the right (1) to propose any claim construction American considers appropriate, (2) to contest any claim construction proposed by IV that American considers inappropriate or inaccurate, and/or (3) to take positions with respect to claim construction issues that are inconsistent with, or even contradictory to, claim construction positions expressed or implied in these Preliminary Invalidation Contentions.

Prior art not included in these Preliminary Invalidation Contentions, whether now known to American, might become relevant depending on the claim constructions proposed by American and/or the Court's claim construction rulings. American reserves all rights to supplement or modify the positions and information in these Preliminary Invalidation Contentions, including without limitation the prior art and grounds of invalidity set forth herein, pursuant to P.R. 3-6 after the Court has construed the Asserted Claims.

D. Ongoing Discovery and Supplementation

American's investigation, including its investigation of prior art and grounds for invalidity, is ongoing. Furthermore, American's invalidity positions will be the subject of expert testimony. American bases these Preliminary Invalidation Contentions on its current knowledge

and understanding of the Asserted Claims, IV's Infringement Contentions, the prior art, systems, and other facts and information available as of the date of these contentions.

American reserves the right to supplement these Preliminary Invalidity Contentions, including, without limitation, by adding additional prior art and grounds of invalidity in accordance with the Federal Rules of Civil Procedure, the Local Rules, the Local Patent Rules, the Docket Control Order, any Order issued by this Court, or otherwise.

E. Prior Art Identification and Citations Thereto

In these Preliminary Invalidity Contentions, American identifies specific portions of prior art references that disclose the elements of the Asserted Claims. While American has identified exemplary prior art references for each element, they do not necessarily identify every disclosure of the same element in each prior art reference. A person of ordinary skill in the art would read a prior art reference as a whole and in the context of other publications, literature, and general knowledge in the field and would rely upon other information including other publications and general scientific or engineering knowledge. American therefore reserves the right to rely upon other unidentified portions of the prior art references and on other publications and prior art products and expert testimony to provide context and to aid understanding and interpretation of the identified portions of the prior art.

American also reserves the right to rely upon (1) other portions of the cited prior art references, other publications, prior art products, and the testimony of experts to establish that the alleged inventions would have been obvious to a person of ordinary skill in the art, including on the basis of modifying or combining certain cited references; (2) all versions of a cited prior art publication; (3) admissions relating to prior art in the Asserted Patents or related patents, the prosecution history of the Asserted Patents or related patents, or other admissions obtained

during discovery; and (4) foreign counterparts of any U.S. patents identified in American's Preliminary Invalidation Contentions.

Where American identifies a particular figure in a prior art reference, the identification should be understood to encompass the caption and description of the figure as well as any text relating to the figure in the remainder of the prior art reference (e.g., for patent references, text in the specification and prosecution history). Similarly, where an identified portion of text refers to a figure or other material, the identification of the text should be understood to include the referenced figure or other material.

All related prior art references that are themselves subparts or related documents of a broader set of documents should be considered a single prior art reference, as that is how a person of ordinary skill in the art would consider such related references.

F. Invalidity, Unenforceability and/or Ineligibility Based On Non-Required Disclosure(s)

American reserves the right to prove the invalidity, unenforceability and/or ineligibility of one or more of the Asserted Claims on bases other than those required to be disclosed in these disclosures pursuant to P.R. 3-3.

G. No Patentable Weight

American reserves the right to argue that various portions of the Asserted Claims, such as an intended use or result, non-functional descriptive material, and certain preamble language, are entitled to no patentable weight. Mapping of a portion of an Asserted Claim to a prior art reference does not represent that such portion of the claim is entitled to patentable weight when comparing the claimed subject matter to the prior art.

II. INVALIDITY CONTENTIONS

As explained herein and in Exhibits A-F, American contends that each of the Asserted Claims is invalid under 35 U.S.C. §§ 101, 102, 103, and/or 112.

A. P.R. 3-3(a) – Identification of Prior Art

Pursuant to P. R. 3-3(a), and subject to American's reservation of rights as stated herein, American identifies the prior art that anticipates or renders obvious the Asserted Claims in the tables set forth below.⁵ On information and belief, each listed reference qualifies as prior art to the Asserted Patents.

To the extent that any of the following are prior art, American reserves the right to rely upon foreign counterparts of the U.S. patents identified herein; U.S. counterparts of foreign patents and foreign patent applications identified herein; and U.S. and foreign patents and patent applications corresponding to articles and publications identified herein. American also reserves the right to rely upon parent or provisional patents or ancestor patents or patent applications from which any of the patents or patent applications identified herein claim priority to as continuation, divisional, or continuation-in-part applications. Identifications of dates of publication are made based on currently available information and American reserves the right to rely upon an earlier date should evidence supporting an earlier date be discovered.

⁵ American also hereby identifies any systems or products that embody the technology described in any patent or publication identified in these Preliminary Invalidation Contentions. American reserves the right to rely on any documents or other evidence regarding any such systems.

1. Identification of Prior Art

a) '582 patent⁶

Patent/Publication No./Document Name	Date of Filing/Issue/Publication ⁷
U.S. Patent No. 6,304,866 (“Chow”)	Filed – June 27, 1997 Issued – October 16, 2001
U.S. Patent Application Publication No. 2002/0019844 (“Kurowski”)	Filed – January 12, 2001 Published – February 14, 2002
U.S. Patent No. 6,330,583 (“Reiffin”)	Filed – September 9, 1994 Issued – December 11, 2001
U.S. Patent No. 6,212,617 (“Hardwick”)	Filed – June 30, 1998 Issued – April 3, 2001
U.S. Patent Application Publication No. 2002/0103889 (“Markson”)	Filed – June 19, 2001 Published – August 1, 2002
U.S. Patent Application Publication No. 2002/0091786 (“Yamaguchi”)	Filed – November 1, 2001 Published – July 11, 2002
U.S. Patent No. 5,765,146 (“Wolf”)	Filed – November 4, 1993 Issued – June 9, 1998
EP Patent Application Publication No. EP0377993 (“Lorie”)	Filed – December 20, 1989 Published – July 18, 1990
P. Maheshwari, “Improving task scheduling for larger grain execution of parallel functional programs,” [1993] Proceedings of the Twenty-sixth Hawaii International Conference on System Sciences, Wailea, HI, USA, 1993, pp. 594-602 vol.2, doi: 10.1109/HICSS.1993.284066. (“Maheshwari”)	January 8, 1993
“Analysis of First-Come-First-Serve Parallel Job Scheduling,” Uwe Schwiegelshohn and Ramin Yahyapour. Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, pages 629-638 (1998). Available at https://dl.acm.org/doi/10.5555/314613.315031 (“Schwiegelshohn”)	January 1, 1998
Gamma System - University of Wisconsin-Madison	At least as early as 1990
Volcano Parallel Database System - University of Wisconsin-Madison	At least as early as 1990

⁶ Defendant notes that there is a pending IPR (IPR2025-00785) asserting that all claims of the '582 patent are invalid. Defendant hereby incorporates by reference, in full, the Petition and the Exhibits in IPR2025-00785.

⁷ For any patent that claims the benefit of a provisional application(s), American may rely on the provisional application(s) for purposes of establishing invalidity. American may also rely upon the date of filing of any issued patent or patent application to show the priority date of that reference.

Patent/Publication No./Document Name	Date of Filing/Issue/Publication ⁷
U.S. 6,334,137 (“Iida”)	Filed – August 24, 1999 Issued – December 25, 2001
U.S. 2002/0099716 (“Thompson”)	Filed – January 25, 2001 Published – July 25, 2002
JP 20003/30959 (“Aiba”)	Filed – May 18, 1999 Issued – May 24, 2006
U.S. 2002/0040381 (“Steiger”)	Filed – September 28, 2001 Published – April 4, 2002
U.S. 5,325,493 (“Herrell”)	Filed – March 30, 1993 Issued – June 28, 1994
U.S. 5,535,393 (“Reeve”)	Filed – June 5, 1995 Issued – July 9, 1996
U.S. 5,765,166 (“Gotfried”)	Filed – April 23, 1996 Issued – June 9, 1998
S. Kuo, M. Winslett, Y. Chen, Y. Cho, M. Subramaniam and K. Seamons, “Parallel input/output with heterogeneous disks,” Proceedings. Ninth International Conference on Scientific and Statistical Database Management (Cat. No.97TB100150), Olympia, WA, USA, 1997, pp. 79-90, doi: 10.1109/SSDM.1997.621154.	Published Aug. 11-13, 1997 Retrieved from https://ieeexplore.ieee.org/document/621154
U.S. 5,945,990 (“Morrison”)	Filed – November 19, 1996 Issued – August 31, 1999
WO 03/012696 (“Smith”)	Filed – July 29, 2002 Published – February 13, 2003
U.S. 6,480,876 (“Rehg”)	Filed – May 28, 1998 Issued – November 12, 2002
“Some Thoughts on Parallel Processing,” Lynn D. Yarbough, Communications of the ACM, Vol. 3, Issue 10, page 539, October 1, 1960 (available at: https://dl.acm.org/doi/10.1145/367415.367426).	Oct. 1, 1960
“A Survey of Some Theoretical Aspects of Multiprocessing,” J. L. Baer, ACM Computing Surveys (CSUR), Volume 5, Issue 1 Pages 31 – 80, March 1, 1973 (available at: https://dl.acm.org/doi/10.1145/356612.356615)	March 1, 1973
“Parallel Sorting Algorithms for Tightly Coupled Multiprocessors,” Michael J. Quinn, Parallel Computing 6 (1988) 349-357.	1988
“Removing Skew Effect in Join Operation on Parallel Processors,” Ron-Chung Hu, Richard R. Muntz, Computer Science Technical Report, University of California, Los Angeles, June 1989.	June 1989
“Parallel Architectures for Database Systems,” A.R.	1989

Patent/Publication No./Document Name	Date of Filing/Issue/Publication ⁷
Hurson, L.L. Miller, S.H. Pakzad, M.H. Eich, B. Shirazi, Advances in Computers, Vol. 28, pages 107-151, 1989.	
“Parallel Algorithms for Merging and Sorting,” Narsingh Deo, Dilip SarKar, Information Sciences, Vol. 56, pages 151-161, 1991.	1991
“Parallel Algorithms for least Median of Squares Regression,” Chong-Wei Xu and Wei-Kei Shiue, Computational Statistics & Data Analysis, Vol. 16, pages 349-362, 1993.	1993
“On the Average Running Time of Odd-Even Merge Sort,” Christine Rub, Max Plank-Institute Fur Informtik, Research Report, April 1995.	April 1995
“Parallel Algorithms,” Guy E. Blelloch and Bruce M. Maggs, a chapter in “The Computer Science and Engineering Handbook.” CRC Press, 1997, ISBN: 0-8493-2909-4. Available at https://dl.acm.org/doi/pdf/10.5555/1882723.1882748 .	1997
“An Approach to Parallelizing Isotonic Regression,” Anthony J. Kearsley, Richard A. Tapia, Michael W. Trosset, Applied Mathematics and Parallel Computing, H. Fisher et. al. (eds.), Physica-Verlag Heidelberg (1996), pages 141-147.	1996
“Part 1: The Parallel Computing Environment, Alice E. Koniges, Morris A. Jette, and David C. Eder, in “Industrial Strength Parallel Computing,” Morgan Kaufmann, (2000). ISBN 9781558605404. Available at: http://wayback.cecm.sfu.ca/PSG/book/intro.html and at https://books.google.com/books?id=mWalawBciCQC&pg=PA1&source=gbs_toc_r&cad=2#v=onepage&q&f=false .	2000
GPFS: A Parallel File System, Jason Barkes, Marcelo R. Barrios, Francis Cougard, Paul G. Crumley, Didac Marin, Hari Reddy, Theerapong Thitayanun, IBM International Technical Support Organization, April 1998.	April 1998
“Analysis of First-Come-First-Serve Parallel Job Scheduling,” Uwe Schwiegelshohn and Ramin Yahyapour. Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (1998). Available at http://dx.doi.org/10.1145/314613.315031 .	1998
“Improving First-Come-First-Serve-Job Scheduling by Gang Scheduling,” Uwe Schwiegelshohn and Ramin Yahyapour. Job Scheduling Strategies for	March 30, 1998

Patent/Publication No./Document Name	Date of Filing/Issue/Publication ⁷
Parallel Processing, IPPS/SPDP '98 Workshop, Orlando, Florida, USA, March 30, 1998, pp. 180-198, available at https://www.academia.edu/14780669/Improving_first_come_first_serve_job_scheduling_by_gang_scheduling	
Bricker, et al., "Condor Technical Summary," University of Wisconsin – Madison Computer Sciences Department	Jan. 1992
W. Gentsch, "Sun Grid Engine: Towards Creating a Compute Power Grid," Sun Microsystems, Inc.	2001
O. Sievert, "A Gentle Introduction to Globus," Prepared for Parallel Computation, CSE160/CSE260, Spring 1999	1999
Anderson, et al., "SETI@home Internet Distributed Computing for SETI," ASP Conference Series, Vol. 213	2000
Bamford, et al., "Architecture of Oracle Parallel Server," Proceedings of the 24 th Very Large Database (VLDB) Conference New York, USA, 1998	1998
J. Howard, "An Overview of the Andrew File System," Carnegie Mellon University	1988
S. Hotovy, "Workload Evolution on the Cornell Theory Center IBM SP2" Cornell Theory Center	1996
Foster et al., "Wide-Area Implementation of the Message Passing Interface," Parallel Computing 24(12),1735-1749	1998
Heymann et al., "Adaptive Scheduling for Master-Worker Applications on the Computational Grid," Department of Computer Sciences University of Wisconsin – Madison	2000
D. Gelernter and N. Carriero, "How to Write Parallel Programs: A Guide to the Perplexed," <i>ACM Computing Surveys</i> , Vol. 21, No. 3	Sept. 1989

b) '722 patent⁸

Patent/Publication No./Document Name	Date of Filing/Issue/Publication
U.S. Patent No. 5,721,825 (“Lawson”)	Filed – October 3, 1996 Issued – February 24, 1998
U.S. Patent No. 6,480,883 (“Tsutsumitake”)	Filed – June 29, 1999 Issued – November 12, 2002
PCT Patent Application Publication No. WO 99/26127 (“Kostes”)	Filed – November 13, 1998 Published – May 27, 1999
U.S. Patent No. 6,999,991 (“Ikeda”)	Filed – July 24, 2000 Issued – February 14, 2006
U.S. Patent No. 6,990,591 (“Pearson”)	Filed – December 22, 1999 Issued – January 24, 2006
European Patent Application Publication No. EP1043671 (“Bird”)	Filed – March 17, 2000 Published – October 11, 2000
U.S. Patent No. 6,721,740 (“Skinner”)	Filed – June 5, 1998 Issued – April 13, 2004
U.S. Patent No. 6,710,702 (“Averbuch”)	Filed – November 22, 1999 Issued – March 23, 2004
U.S. Patent No. 6,763,384 (“Gupta”)	Filed – July 10, 2000 Issued – July 13, 2004
U.S. Patent No. 6,859,829 (“Parupudi”)	Filed – February 23, 1999 Issued – February 22, 2005
U.S. Patent No. 6,216,132 (“Chandra”)	Filed – November 20, 1997 Issued – April 10, 2001
U.S. Patent No. 6,910,070 (“Mishra”)	Filed – January 24, 2000 Issued – June 21, 2005
U.S. Patent No. 5,754,939 (“Herz”)	Filed – October 31, 1995 Issued – May 19, 1998
U.S. Patent No. 6,823,359 (“Heidingsfeld”)	Filed – November 21, 2000 Issued – November 23, 2004
U.S. Patent No. 6,757,283 (“Yamanaka”)	Filed – January 24, 2000 Issued – June 29, 2004
U.S. Patent No. 7,051,084 (“Hayton”)	Filed – November 2, 2000 Issued – May 23, 2006
U.S. Patent No. 6,633,910 (“Rajan”)	Filed – December 14, 1999 Issued – October 14, 2003
U.S. Patent No. 9,300,560 (“Leighton”)	Filed – March 4, 2013

⁸ American notes that there is a pending IPR asserting that all claims of the '722 patent are invalid, IPR2025-00987, and a settled IPR, IPR2025-00200. American hereby incorporates by reference, in full, the Petition and the Exhibits in both IPR2025-00987 and IPR2025-00200. American further incorporates by reference any post grant review proceeding (including any IPRs or EPRs) that has been or may be filed challenging the patentability of one or more Asserted Claim.

Patent/Publication No./Document Name	Date of Filing/Issue/Publication
	Issued – March 29, 2016
U.S. Patent No. 5,951,694 (“Choquier”)	Filed – February 3, 1997 Issued – September 14, 1999
PointCast System – PointCast Incorporated	At least as early as 1997
Castanet System – Marimba, Inc.	At least as early as 1996
Active Desktop System – Microsoft Corporation	At least as early as 1998
U.S. 2002/0032722 (“Baynes”)	Filed – September 12, 2001 Published – March 14, 2002
U.S. 7,103,680 (“Holdsworth”)	Filed – April 25, 2000 Issued – September 5, 2006
U.S. 6,886,044 (“Miles”)	Filed – June 30, 1999 Issued – April 26, 2005
U.S. 6,654,786 (“Fox”)	Filed – August 30, 2000 Issued – November 25, 2003
U.S. 5,913,032 (“Schwartz”)	Filed – September 30, 1996 Issued – June 15, 1999
U.S. 7,516,196 (“Madan”)	Filed – March 21, 2000 Issued – April 7, 2009
U.S. 2010/0325562 (“Andrews”)	Filed – August 27, 2010 Published – December 23, 2010
U.S. 6,643,682 (“Todd”)	Filed – February 22, 2000 Issued – November 4, 2003
JP JPH11353216A (“Masaru”)	Filed – June 8, 1998 Published – December 24, 1999
U.S. 6,694,352 (“Omoigui”)	Filed – December 16, 1999 Issued – February 17, 2004
U.S. 6,336,119 (“Banavar”)	Filed – March 30, 2000 Issued – January 1, 2002
U.S. 5,978,842 (“Noble”)	Filed – July 18, 1997 Issued – November 2, 1999
U.S. 6,760,759 (“Chan”)	Filed – May 1, 2000 Issued – July 6, 2004
GB2348025 (“Bird GB”)	Filed – March 19, 1999 Published – May 12, 1999
M. Hauswirth and M. Jazayeri, “A Component and Communication Model for Push Systems,” Proceedings of the ESEC/FSE 99 - Joint 7th European Software Engineering Conference (“Hauswirth”)	Pub. at least as early as September 6, 1999.
G. S. Barnes Nelson, “Messaging Systems: SAS® Tools for Internet-Based Communications,” Proceedings of the 25th Annual SAS Users Group International Conference (SUGI 25), Paper 288-25 (“Barnes Nelson”)	Pub. at least as early as March 19, 1999.
Segall, et al., “Content Based Routing with Elvin4,”	Pub. at least as early as June 2000.

Patent/Publication No./Document Name	Date of Filing/Issue/Publication
Proceedings of AUUG2K (No. 39) (“Segall”)	

c) ’785 patent

Patent/Publication No./Document Name	Date of Filing/Issue/Publication
U.S. Patent No. 6,970,941 (“Caronni I”)	Filed – December 10, 1999 Issued – November 29, 2005
U.S. Patent No. 7,814,228 (“Caronni II”)	Filed – February 13, 2003 Issued – October 12, 2010
U.S. Patent No. 6,766,371 (“Hipp”)	Filed – October 5, 2000 Issued – July 20, 2004
U.S. Patent Application Publication No. 2003/0028671 (“Mehta”)	Filed – June 10, 2002 Published – February 6, 2003
Huitema, C., Network Working Group Request for Comment (RFC): 1383, entitled An Experiment in DNS Based IP Routing (“RFC 1383”)	Published December 1992
U.S. Patent No. 7,133,404 (“Alkhatib”)	Filed – August 10, 2001 Issued – November 7, 2006
U.S. Patent Application Publication No. 2002/0184529 (“Foster I”)	Filed – April 19, 2002 Published – December 5, 2002
U.S. Patent Application Publication No. 2003/0208602 (“Bhalla”)	Filed – April 8, 2002 Published – November 6, 2003
U.S. Patent Application Publication No. 2002/0181395 (“Foster II”)	Filed – April 19, 2002 Published – December 5, 2002
U.S. Patent Application Publication No. 2002/0091859 (“Tuomenoksa”)	Filed – April 11, 2001 Published – July 11, 2002
U.S. Patent Application Publication No. 2004/0177136 (“Chen”)	Filed – March 3, 2003 Published – September 9, 2004
European Patent Publication No. EP1098496 (“Engmann”)	Published – October 19, 2000
U.S. Patent Application Publication No. 2002/0069278 (“Forslow”)	Filed – December 5, 2000 Published – June 6, 2002
U.S. Patent No. 6,701,437 (“Hoke”)	Filed – November 9, 1998 Issued – March 2, 2004
U.S. Patent Application Publication No. 2002/0116502 (“Iyer”)	Filed – February 22, 2001 Published – August 22, 2002
U.S. Patent Application Publication No. 2003/0065785 (“Jain”)	Filed – September 28, 2001 Published – April 3, 2003
U.S. Patent Application Publication No. 2002/0035624 (“Kim”)	Filed – July 9, 2001 Published – March 21, 2002
U.S. Patent Application Publication No. 20020154635 (“Liu”)	Filed – April 23, 2001 Published – October 24, 2002
U.S. Patent No. 8,068,817 (“Viswanath”)	Filed – August 27, 2002 Issued – November 29, 2011

U.S. Patent Application Publication No. 2002/0116523 (“Warrier”)	Filed – February 22, 2001 Published – August 22, 2002
U.S. Patent Application Publication No. 2003/0229697 (“Borella”)	Filed – June 10, 2002 Published – December 11, 2003
U.S. Patent Application Publication No. 2003/0084162 (“Johnson”)	Filed – October 31, 2001 Published – May 1, 2003
PCT Patent Application Publication No. WO2003088625 (“Volz”)	Filed – March 28, 2003 Priority – April 8, 2002 Published – October 23, 2003
U.S. Patent Application Publication No. 2004/0098507 (“Thubert”)	Filed – November 20, 2002 Published – May 20, 2004
U.S. Patent Application Publication No. 2003/0016636 (“Tari”)	Filed – July 15, 2002 Published – January 23, 2003
U.S. Patent No. 8,107,483 (“Dunk”)	Filed – March 12, 2010 Priority – November 27, 2002 Issued – January 31, 2012
PCT Patent Application Publication No. WO2002015014 (“Wootton”)	Filed – July 31, 2001 Published – February 21, 2002
U.S. Patent Application Publication No. 2002/0038382 (“Ryu”)	Filed – August 30, 2001 Published – March 28, 2002
U.S. Patent Application Publication No. 2003/0172184 (“Kong”)	Filed – March 6, 2003 Published – September 11, 2003
U.S. Patent Application Publication No. 2004/0037260 (“Kakemizu”)	Filed – August 7, 2003 Published – February 26, 2004
Cisco VPN Routers and Software Systems – Cisco Systems	At least as early as 2002
VMware Systems - VMware	At least as early as 2001
U.S. Patent Application Publication No. 2002/0029276 (“Bendinelli”)	Filed – April 11, 2001 Published – October 7, 2003
“Virtual Enterprise Networks: The Next Generation of Secure Enterprise Networking,” Proceedings of the Sixteenth Annual Computer Security Applications Conference (ACSAC'00) (“NextGen”)	Date of Conference: December 11-15, 2000 Published on IEEE Xplore: August 6, 2002

d) '844 patent

Patent/Publication No./Document Name	Date of Filing/Issue/Publication
“Optimizing the Migration of Virtual Computers,” Association for Computing Machinery, Special Interest Group on Operating Systems, Operating Systems Review, Proceedings of the 5th Symposium on Operating Systems Design and Implementation, Volume 36, Issue S1 (“Sapuntzakis”)	Published December 31, 2002
U.S. Patent No. 7,089,300 (“Birse”)	Filed – October 19, 1999 Issued – August 8, 2006

U.S. Patent Application Publication No. 2005/0283597 (“Holzmann”)	Filed – June 22, 2004 Published – December 22, 2005
U.S. Patent Application Publication No. 2002/0034105 (“Kulkarni”)	Filed – January 8, 2001 Published – March 21, 2002
U.S. Patent Application Publication No. 2005/0004925 (“Stahl”)	Filed – May 7, 2004 Published – January 6, 2005
U.S. Patent Application Publication No. 2006/0168154 (“Zhang”)	Filed – November 19, 2004 Published – July 27, 2006
U.S. Patent No. 8,001,085 (“Kiselev”)	Filed – November 25, 2003 Issued – August 16, 2011
U.S. Patent Application Publication No. 2004/0172509 (“Takeda”)	Filed – June 23, 2003 Published – September 2, 2004
U.S. Patent No. 6,671,782 (“Menon”)	Filed – March 29, 2000 Issued – December 30, 2003
U.S. Patent Application Publication No. 2003/0236850 (“Kodama”)	Filed – March 21, 2002 Published – December 25, 2003
EP Patent No. EP0604013 (“Nelson”)	Filed – November 15, 1993 Issued – February 20, 2002
PCT Application Publication No. WO 2006/074869 (“Bayer”)	Filed – January 4, 2006 Published – July 20, 2006
Shrira, Liuba et al., “Thresher: an efficient storage manager for copy-on-write snapshots,” ATEC ‘06: Proceedings of the annual conference on USENIX ‘06 Annual Technical Conference (Shrira)	Published at least by May 30, 2006
Laros, James H. et al., “Implementing Scalable Disk-Less Clusters Using The Network File System (NFS),” (“Laros”) Retrieved from: https://www.researchgate.net/publication/255223070_Implementing_scalable_disk-less_clusters_using_the_Network_File_System_NFS	Published at least by January 2003
U.S. Patent Application Publication No. 2004/0153639 (“Cherian”)	Filed – February 5, 2003 Published – August 5, 2004
U.S. Patent Application Publication No. 2007/0057958 (“Bucher”)	Filed – August 9, 2006 Published – March 15, 2007
U.S. Patent No. 7,197,608 (“Mikuma”)	Filed – July 12, 2004 Issued – March 27, 2007
U.S. Patent Application Publication No. 2005/0283575 (“Kobayashi”)	Filed – August 13, 2004 Published – December 22, 2005
U.S. Patent No. 6,453,334 (“Vinson”)	Filed – June 16, 1998 Issued – September 17, 2002
U.S. Patent No. 5,930,513 (“Taylor”)	Filed – June 6, 1996 Issued – July 27, 1999

U.S. Patent Application Publication No. 2002/0002660 (“Malcolm”)	Filed – July 31, 1998 Published – January 3, 2002
U.S. Patent Application Publication No. 2002/0091763 (“Shah”)	Filed – May 15, 2001 Published – July 11, 2002
U.S. Patent No. 7,398,382 (“Rothman”)	Filed – December 29, 2004 Issued – July 8, 2008
U.S. Patent No. 6,317,826 (“McCall”)	Filed – October 19, 1998 Issued – November 13, 2001
Southwest Airlines NetApp System – NetApp/Southwest	At least as early as 2002
VMware Systems - VMware	At least as early as 2001
U.S. Patent No. 6,618,736 (“Menage”)	Filed – March 9, 2001 Issued – September 9, 2003
Apple NetBoot	At least as early as 2002
U.S. Patent Application Publication No. 2002/0186698 (“Ceniza”)	Filed – June 12, 2001 Published – December 12, 2002

e) ‘469 patent⁹

Patent/Publication No./Document Name	Date of Filing/Issue/Publication
U.S. Patent No. 6,894,990 (“Agarwal”)	Filed – October 13, 2000 Issued – May 17, 2005
U.S. Patent Application Publication No. 2002/0138625 (“Bruner”)	Filed – March 21, 2001 Published – September 26, 2002
U.S. Patent No. 6,445,777 (“Clark”)	Filed – December 12, 1998 Issued – September 3, 2002
DeSanctis, et al., “Aeronautical Communications for Personal and Multimedia Services via Satellite,” AM. INST. OF AERONAUTICAL. AND ASTRONAUTICS (“DeSanctis”)	Published at least by April 2003
U.S. Patent No. 6,968,394 (“El-Rafie”)	Filed – March 21, 2000 Published – November 22, 2005
U.S. Patent Application No. 2002/0075844 (“Hagen”)	Filed – April 10, 2001 Published -- June 20, 2002
Jahn, et al., “Dimensioning of Aeronautical Satellite Services,” AM. INST. OF AERONAUTICAL. AND ASTRONAUTICS (“Jahn I”)	Published at least by October 2002

⁹ American notes that there is a pending IPR (IPR2025-00782) asserting that the asserted claims of the ‘469 patent are invalid. American hereby incorporates by reference, in full, the Petition and the Exhibits in IPR2025-00782.

Jahn, et al., “Passenger Multimedia Service Concept via Future Satellite Systems” DGLR JAHRBUCH (“Jahn II”)	2002
U.S. Patent Application No. 2003/0051041 (“Kalavade”)	Filed – August 6, 2002 Published – March 13, 2003
Korean Patent No. 2003/0029267 (“Lee”)	Filed – October 5, 2001 Published – April 14, 2003
U.S. Patent Application No. 2007/0008937 (“Mody”)	Filed – July 21, 2004 Published – January 11, 2007
U.S. Patent Application No. 2003/0055975 (“Nelson”)	Filed – June 19, 2001 Published – March 28, 2003
U.S. Patent No. 6,377,981 (“Ollikainen”)	Filed November 20, 1997 Published – April 23, 2002
Czech Patent No. 2,000,578 (“Rothblatt I”)	Filed – August 19, 1998 Published – August 16, 2000
U.S. Patent No. 6,105,060 (“Rothblatt II”)	Filed – September 5, 1997 Published – August 15, 2000
U.S. Patent Application No. 2004/0109449 (“Seo”)	Filed – August 15, 2003 Published – June 10, 2004
Suwanateep, Manasuda, “Inflight Internet Connectivity Technology and Its Application on E-Commerce,” U. OF THAILAND (2002) (“Suwanateep”)	Published at least by July 2002
U.S. Patent No. 5,987,430 (“Van Horne”)	Filed – August 28, 1997 Published – November 16, 1999
U.S. Patent Application Publication No. 2003/0050041 (“Wu”)	Filed – September 7, 2001 Published – March 13, 2003

f) '326 patent¹⁰

Patent/Publication No./Document Name	Date of Issue/Publication/Priority
PCT Patent App. Pub. No. 2000/038387 (“Baldemair”)	Filed – December 21, 1999 Published – June 29, 2000
U.S. Patent No. 7,418,050 (“Gardner”)	Filed – February 26, 2003 Issued – August 26, 2008
U.S. Patent App. Pub. No. 2003/0022473 (“Yamaura”)	Filed – June 17, 2002 Published – January 30, 2003
U.S. Patent No. 6,510,133 (“Uesugi”)	Filed – May 28, 1998 Issued – January 21, 2003
JP Patent App. Pub. No. JP2002319917 (“Mori”)	Filed – April 24, 2001 Published – October 31, 2002

¹⁰ American notes that there is a pending IPR (IPR2025-01055) asserting that the asserted claims of the '326 patent are invalid. American hereby incorporates by reference, in full, the Petition and the Exhibits in IPR2025-01055.

JP Patent App. Pub. No. JP2003309533 (“Miyoshi”)	Filed – April 17, 2002 Published – October 31, 2003
JP Patent App. Pub. No. JPH11113049 (“Koga”)	Filed – September 30, 1997 Published – April 23, 1999
Thompson, D. E., “Modelling Adjacent Channel Interference in 3G Networks,” THE INSTITUTE OF ELECTRICAL ENGINEERS (2003) (“Thompson”)	2003
Weste, Neil et al., “VLSI for OFDM,” IEEE COMMUNICATIONS MAGAZINE (Oct. 1998) (“Weste”)	1998
“WLAN Channel Bonding: Causing Greater Problems Than It Solves” (“Texas Instruments White Paper”)	Sep. 2003
U.S. Patent No. 7,095,708 (“Alamouti”)	Filed – June 14, 2000 Issued – August 22, 2006
U.S. Patent No. 6,975,585 (“Olafsson”)	Filed – July 27, 2000 Issued – December 13, 2005

1. Identification Of Prior Art Sales/Public Uses

Item Offered for Sale and/or Publicly Used	Date of Offer/Public Use	Person or Entity Who Made and Received Offer, Made Public Use, or Made Information Known
NetApp’s WAFL File System		This architecture was thoroughly documented in both U.S. Patents 5,819,292 and 7,334,095, as well as in technical literature such as “File System Design for an NFS File Server Appliance” by Dave Hitz et al.
VMware’s GSX and ESX server platforms		
Sun Microsystems’ diskless workstations		
The Episode File System	Early 1990s	
Petal		HP Labs
“WLAN Solutions: TNETW1130 Converged Single-Chip MAC and Baseband Processor for IEEE 802.11 a/b/g” (“TI Baseband Processor for IEEE 802.11”)	2003	Texas Instruments, Inc.
Airpath Wireless Hot Spot	No later than June 2003	Airpath Wireless, Inc., Mainstream Data, Inc., Todd Meyers
Connexion	No later than June 2003	The Boeing Co., Connexion by Boeing, Scott E. Carson, Cisco Systems, Inc., American Airlines, Inc., Delta Air Lines, Inc., United

Item Offered for Sale and/or Publicly Used	Date of Offer/Public Use	Person or Entity Who Made and Received Offer, Made Public Use, or Made Information Known
		Airlines, Inc., Lufthansa Group
Maritime Telecommunications Network Cruise Ship Internet Services	No later than June 2003	MTN Satellite Communications (MTN), formerly known as Maritime Telecommunications Network, PCTEL, Inc., Digital Seas International (DSI), Norwegian Cruise Line (NCL), Semester at Sea (SaS), and other cruise line customers of MTN
Stratos	No later than September 2003	Stratos Global Corporation; Inmarsat Global Ltd.
Tenzing Global	No later than April 2001	Tenzing Communications, Inc.; Tenzing Technologies, LLC; Airbus SE; Telia company AB, SITA, OnAir company, Cathay Pacific Airways Ltd., Singapore Airlines Ltd., Virgin Atlantic Airways Ltd.
Wireless Cabin	No later than February 2003	Inmarsat Global Ltd., Telefonaktiebolaget LM Ericsson, Siemens AG, Kid-Systeme GmbH, ESYS Consulting Ltd., Airbus SE, University of Bradford

B. P.R. 3-3(b) – Anticipation and Obviousness

Pursuant to P.R. 3-3(b), and subject to American’s reservation of rights, American attaches claim charts hereto that are directed to the prior art references that anticipate each of the Asserted Claims under 35 U.S.C. §§ 102(a), (b), (e), and/or (g), either expressly or inherently, and/or the prior art references that, in the alternative, would have rendered the Asserted Claims obvious under 35 U.S.C. § 103. *See* Exhibits A-F. Some claim charts also contain explanations about the motivation to combine the references. The combinations contained in American’s claim charts are exemplary. Any prior art reference cited herein may be combined with any other reference to demonstrate the invalidity of any of the Asserted Claims, as set forth below and in Exhibits A-F.

To the extent any claim limitation is construed to have a similar meaning, or to encompass similar feature(s) and/or function(s), as any other claim limitation, the citations to prior art references for each of those claim limitations in American's claim charts are incorporated by reference with respect to each other.

American's claim charts provide exemplary citations to the prior art references that teach or suggest every element of the Asserted Claims. To the extent that an element of an Asserted Claim is not shown in a chart, the Asserted Claims would have been obvious based on a combination of one or more other prior art references, as set forth below and in Exhibits A-F.

Much of the art cited in these Preliminary Invalidity Contentions reflects common knowledge and the state of the art at the time of the earliest filing date of the Asserted Patents. American may rely on additional citations, references, expert testimony, and other material to provide context or to aid in understanding the cited portions of the references and/or cited features of the systems. American also may rely on expert testimony explaining relevant portions of references, relevant hardware or software products or systems, and other discovery regarding these subject matters. Additionally, American may rely on other portions of any prior art reference or other references relied on by the same authors or describing the same systems for purposes of explaining the background and general technical subject area of the reference.

Where an individual reference is cited with respect to all elements of an Asserted Claim, American contends that the reference anticipates the claim under 35 U.S.C. §§ 102(a), (b), (e), and/or (g) and also renders obvious the claim under 35 U.S.C. § 103, both by itself in view of the knowledge of a person of ordinary skill in the art and in combination with the other cited references to the extent the reference is not found to disclose one or more claim elements. A single prior art reference, for example, can establish obviousness where the differences between

the disclosures within the reference and the claimed invention would have been obvious to one of ordinary skill in the art. For example, “[c]ombining two embodiments disclosed adjacent to each other in a prior art patent does not require a leap of inventiveness.” *Boston Scientific Scimed, Inc. v. Cordis Corp.*, 554 F.3d 982, 991 (Fed. Cir. 2009). To the extent IV contends that an embodiment within a particular item of prior art does not fully disclose all limitations of a claim, American accordingly reserves its rights to rely on other embodiments in that prior art reference, or other information, to show single reference obviousness under 35 U.S.C. § 103(a).

Where an individual reference is cited with respect to fewer than all elements of an Asserted Claim, American contends that the reference renders obvious the claim under 35 U.S.C. § 103(a) in view of each other reference and combination of references that discloses the remaining claim element(s), as indicated in the claim charts submitted herewith. “Under § 103, the scope and content of the prior art are to be determined; differences between the prior art and the claims at issue are to be ascertained; and the level of ordinary skill in the pertinent art resolved. Against this background, the obviousness or nonobviousness of the subject matter is determined.” *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007) (quoting *Graham v. John Deere Co.*, 383 U.S. 1, 17 (1966)).

Exemplary motivations to combine references are discussed in the accompanying charts. American reserves the right to rely upon any references or assertions identified herein in connection with American’s contentions that each Asserted Claim is invalid under 35 U.S.C. § 103 and to rely upon expert testimony addressing such references and assertions. The fact that prior art is identified to anticipate the Asserted Claims presents no obstacle in also relying on that reference as a basis for invalidity based on obviousness. It is established that “a rejection for obviousness under § 103 can be based on a reference which happens to anticipate the claimed

subject matter.” *In re Meyer*, 599 F.2d 1026, 1031 (C.C.P.A. 1979). To the extent any cited prior art item may not fully disclose a limitation of an Asserted Claim or is alleged by IV to lack disclosure of the limitation, such limitation is present and identified in another prior art item as shown in the attached claim charts.

Many of the cited references cite or relate to additional references and/or products, services, or projects. Many of the cited references also cite software, hardware, or systems. American might rely upon such cited additional references and/or products and copies or exemplars of such software, hardware, or systems. American will produce or make available for inspection any such cited references, products, software, hardware, or systems that it intends to rely upon. American may also rely upon the disclosures of the references cited and/or discussed during the prosecution of the Asserted Patents and/or the assertions presented regarding those references.

American reserves the right to further streamline and reduce the number of anticipation or obviousness references relied upon with respect to a given Asserted Claim and to exchange or otherwise modify the specific references relied upon for anticipation and within each obviousness combination for each Asserted Claim.

C. P.R. 3-3(c) – Claim Charts

American attaches the following claim charts pursuant to P.R. 3-3(c):

Asserted Patent	Claim Chart Exhibits
'582 patent	A1-A6
'722 patent	B1-B12
'785 patent	C1-C33
'844 patent	D1–D26; OTDP-1–OTDP-2
'469 patent	E1-E18
'326 patent	F1-F13

The attached claim charts are based, in whole or in part, on IV’s asserted theories of infringement in this case, to the extent discernible from IV’s Infringement Contentions. As an

initial matter, all portions of each prior art reference cited in each of the attached claim charts are relied upon to support the disclosure of each patent claim limitation, as all portions provide general support. Representative descriptions and supporting citations are nevertheless provided but are merely exemplary; they do not necessarily reflect every instance where a particular claim term or claim limitation may be disclosed in or taught by the prior art reference. References to figures or drawings refer to the figures/drawings themselves, as well as to any accompanying text or text necessary to understand the figures or drawings. References to text refers to the text itself, as well as the accompanying figures or drawings that accompany the text. American reserves the right to rely on additional, or different, portions of the prior art references, other publications and expert testimony to establish what these references would have taught one of ordinary skill in the art, or in what manner they would have motivated a particular combination of references. Moreover, in certain instances, representative documents for certain prior art systems are cited, but, again, they are merely exemplary; they do not necessarily reflect or include every document relating to the prior art system that exists and that discloses or teaches a particular claim term or claim limitation. American reserves the right to rely on any and all documents that describe or relate to prior art systems, including relying on the system itself. American also reserves the right to rely on the testimony of the authors, named inventors, or anyone else with knowledge of the prior art references and systems identified herein, as well as expert testimony regarding any such references or systems.

D. Obviousness and Motivation to Combine

The primary references identified above, and as further described in Exhibits A-F, each discloses, either expressly or inherently, every element of the Asserted Claims, thereby anticipating those claims. To the extent IV contends that any primary reference does not

anticipate the Asserted Claims, it would have been obvious to combine or modify the primary references with concepts from other prior art, as explained herein and in Exhibits A-F.

In particular, for each limitation of the Asserted Claims that IV contends is not met by a particular primary reference, American contends that the limitation (and claim as a whole) is obvious based on a combination of that particular primary reference with (1) any other primary reference disclosing that limitation, (2) any admitted prior art, as explained in the background of each patent or discussed in the file history, (3) any reference identified in Exhibits A-F as disclosing that limitation, and/or (4) the knowledge of a person of ordinary skill in the art and/or any of the references and concepts discussed herein regarding the relevant background and state of the art. The specific combinations of prior art that American contends render the Asserted Claims obvious are readily determinable as described herein, which is the most efficient manner of identifying the combinations in light of the fact that IV has asserted 21 claims across 6 patents. American's obviousness grounds for each dependent claim incorporate the obviousness grounds for the claim(s) from which the dependent claim depends in addition to any obviousness grounds identified in the charts for the dependent claim.

American does not yet have the benefit of IV's positions on the prior art, including what (if any) elements it contends are missing in each prior art reference, whether IV agrees that a reference is in fact prior art, and whether IV agrees that a person of ordinary skill in the art would be motivated to combine specific references. American reserves the right to supplement these obviousness positions (including identifying additional prior art combinations and the associated reasons to combine) as discovery in the case progresses, including expert discovery.

Each prior art reference may be combined with one or more other prior art references to render obvious the Asserted Claims in combination, as explained in more detail below. The

disclosures of these references also may be combined with information known to persons skilled in the art at the time of the alleged invention and understood and supplemented in view of the common sense of persons skilled in the art at the time of the alleged invention, including any statements in the intrinsic record of the Asserted Patents and related applications.

A person of ordinary skill would have been motivated to combine the prior art cited in the attached claim charts based on the nature of the problem to be solved, the teachings of the prior art, and the knowledge of persons of ordinary skill in the art. The identified prior art references, including portions cited in the claim charts, address the same or similar technical issues and suggest the same or similar solutions to those issues as the Asserted Claims. On such bases, on an element-by-element basis, American intends to combine one or more prior art references identified in the claim charts attached as Exhibits A-F with each other to address any contention from IV that a particular prior art item supposedly lacks one or more elements of an Asserted Claim. In other words, American contends that each charted prior art reference can be combined with other charted prior art references when a particular prior art item lacks or does not explicitly disclose an element or feature of an Asserted Claim.

A motivation, teaching or suggestion to combine the prior art identified exists for each obviousness combination identified in Exhibits A-F. In *KSR International Co. v. Teleflex Inc., et al.*, 550 U.S. 398 (2007), the Supreme Court held that a claimed invention can be obvious even if there is no teaching, suggestion, or motivation for combining the references to produce the claimed invention. *KSR* holds that patents that are based on new combinations of elements or components already known in a technical field may be found obvious. Specifically, the Court in *KSR* rejected a rigid application of the “teaching, suggestion, or motivation [to combine] test.” *Id.* at 418-419. “In determining whether the subject matter of a patent claim is obvious, neither

the particular motivation nor avowed purpose of the patentee controls. What matters is the objective reach of the claim.” *Id.* at 419. “Under the correct analysis, any need or problem known in the field of endeavor at the time of the invention and addressed by the patent can provide a reason for combining the elements in the manner claimed.” *Id.* at 420. The Court noted that “[c]ommon sense teaches, however, that familiar items may have obvious uses beyond their primary purposes, and in many cases a person of ordinary skill will be able to fit the teachings of multiple patents together like pieces of a puzzle.” *Id.* at 420. The Court further stated that “[t]he combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results” *Id.* at 416. “When there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp. If this leads to the anticipated success, it is likely the product not of innovation but of ordinary skill and common sense.” *Id.* at 421.

Based on *KSR*, the USPTO issued a set of Examination Guidelines. *See* Examination Guidelines for Determining Obviousness Under 35 U.S.C. §103 in view of the Supreme Court Decision in *KSR International Co. v. Teleflex, Inc.*, 72 Fed. Reg. 57526 (October 10, 2007). The Guidelines provide a variety of rationales on which to base a finding of obviousness. Those rationales are based on *KSR* and other precedent. The rationales include:

- (A) Combining prior art elements according to known methods to yield predictable results;
- (B) Simple substitution of one known element for another to obtain predictable results;
- (C) Use of known technique to improve similar devices (methods, or products) in the same way;

- (D) Applying a known technique to a known device (method, or product) ready for improvement to yield predictable results;
- (E) “Obvious to try”— choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success;
- (F) Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations would have been predictable to one of ordinary skill in the art;
- (G) Some teaching, suggestion, or motivation in the prior art that would have led one of ordinary skill to modify the prior art reference or to combine prior art reference teachings to arrive at the claimed invention.

Id. at 57529. American contends that one or more of these rationales apply in considering the obviousness of the Asserted Claims. A person of ordinary skill in the art at the time of the claimed invention(s) had reason to combine or modify one or more of the references listed and charted in Exhibits A-F in light of the knowledge of a person of ordinary skill at the time and information in the prior art references cited herein.

Additional evidence establishing that there would have been a motivation to combine the prior art references identified above pursuant to the Scheduling Order includes the interrelated teachings of multiple prior art references; the effects of demands known to the design community or present in the marketplace; the existence of a known problem for which there was an obvious solution encompassed by the Asserted Claims; the existence of a known need or problem in the field of the endeavor at the time of the invention(s); and the background knowledge that would have been possessed by a person having ordinary skill in the art.

Further, the prior art references explicitly or implicitly reference other prior art references, share common authors or inventors, were published in the same journals, presented at the same conferences, were presented as proposals to industry groups, and/or were developed at common companies, schools, or organizations, all of which would motivate one of skill in the art

to combine them. These references are also within the field of the Asserted Patents and are directed to similar subject matter within that field. Additionally, any products, devices, or processes described in the references existed and/or were invented before or during the period in which the claimed inventions were developed, providing further motivation to combine them.

Thus, the motivation to combine the teachings of the prior art references disclosed herein is found in the references themselves and: (1) the nature of the problem being solved, (2) the express, implied and inherent teachings of the prior art, (3) the knowledge of persons of ordinary skill in the art, (4) the fact that the prior art is generally directed toward methods and systems for processing wireless signals, and (5) the predictable results obtained in combining the different elements of the prior art. Additionally, one would be motivated to address at least the alleged problems or achieve the purported objectives identified in the description of the Asserted Patents. *See, e.g.*, the exemplary modifications to combine detailed below. Moreover, the references charted in the Invalidity Contentions recognized and solved these problems.

Any reference or combination of references that anticipates or makes obvious an asserted independent claim also makes obvious any asserted claim dependent on that independent claim because every element of each dependent claim was known by a person of ordinary skill at the time of the alleged invention, and it would have been obvious to combine those known elements with the independent claims at least as a matter of common sense and routine innovation. Accordingly, American contends that each asserted claim would have been obvious not only by the combinations explicitly defined in these contentions, but also by any combination of references that renders obvious an asserted claim.

Numerous prior art references, including those identified above and in the attached claim charts reflect common knowledge and the state of the prior art prior to the priority dates of the

Asserted Patents. As it would be unduly burdensome to create detailed claim charts for the thousands of invalidating combinations, American has provided illustrative examples of such invalidating combinations in the attached claim charts. For at least the reasons described above and below in the examples provided, as well as in the attached claim charts, it would have been obvious to one of ordinary skill in the art to combine any of a number of prior art references, including any combination of those identified in the attached claim charts, to meet the limitations of the Asserted Claims. As such, American's inclusion of exemplary combinations does not preclude them from identifying other invalidating combinations as appropriate.

Secondary considerations of nonobviousness "simply cannot overcome a strong prima facie case of obviousness." *Wyers v. Master Lock Co.*, 616 F.3d 1231, 1246 (Fed. Cir. 2010). For any secondary consideration to be relevant, IV must establish a nexus between the secondary consideration and the claimed inventions. *Ormco Corp. v. Align Tech., Inc.*, 463 F.3d 1299, 1312 (Fed. Cir. 2006).

IV has not provided any evidence of any secondary considerations, much less any nexus between such secondary considerations and the claimed inventions of the Asserted Patents. American contends that no secondary considerations evidence exists that supports the validity of any Asserted Claim and reserve all rights to respond to any argument or evidence presented by IV regarding any alleged secondary considerations of non-obviousness.

1. Statement Regarding Exemplary Obviousness Combinations and Motivation to Combine

The suggested obviousness combinations discussed in the attached claim charts are not to be construed to suggest that any reference included in the combinations is not anticipatory. Further, to the extent that IV contends that any of the anticipatory prior art fails to disclose one or more limitations of the Asserted Claims, American reserves the right to identify other prior art

references that, when combined with the anticipatory prior art, would render the claims obvious despite an allegedly missing limitation. American will further specify the motivations to combine the prior art, including through reliance on expert testimony, at the appropriate later stage of this lawsuit.

Further, American notes that there are pending IPRs asserting that at least the asserted claims of the patents are invalid. American hereby incorporates by reference, in full the Petition and Exhibits in the following IPRs: IPR2025-00785; IPR2025-00782; IPR2025-00786; IPR2025-01055; IPR2025-00987; IPR2025-00840; and IPR2025-00931.

2. Secondary Considerations

A patentee bears the burden of production with respect to evidence of secondary considerations of non-obviousness. *ZUP, LLC v. Nash Mfg., Inc.*, 896 F.3d 1365, 1373 (Fed. Cir. 2018). As of the date of these Preliminary Invalidity Contentions, IV has not yet identified any evidence of secondary considerations. American reserves all rights to further respond to any secondary considerations of non-obviousness raised by IV, including by updating, modifying, and/or adding to these Preliminary Invalidity Contentions. American is not aware of any unexpected results (none is mentioned in the Asserted Patents or their file histories), long felt need, commercial success (or any nexus to any allegedly successful commercial embodiment), or awards for the alleged inventions of the Asserted Patents.

E. P.R. 3-3(d): Invalidity Under 35 U.S.C. § 112

As set forth below, American contends that the Asserted Claims are invalid under 35 U.S.C. § 112 because the claims (1) are indefinite; (2) fail to satisfy the enablement requirement; and/or (3) fail to satisfy the written description requirement.

American's invalidity contentions for the Asserted Claims identified in this section are made in the alternative and do not constitute, and should not be interpreted as, admissions

regarding the construction or scope of the Asserted Claims identified herein or that any of the Asserted Claims identified herein would not have been anticipated and/or obvious in light of the prior art.

IV has not yet provided a claim construction for any of the terms or phrases that American anticipates will be in dispute. American, therefore, cannot provide a complete list of § 112 defenses because American does not know whether IV will proffer a construction for certain terms and phrases that is broader than, or inconsistent with, the construction that would be supportable by the disclosure set forth in the specification. American offers these contentions without prejudice to any position they may ultimately take as to any claim construction issues.

Accordingly, American reserves the right to supplement, amend, and/or modify these § 112 invalidity contentions as discovery progresses.

1. Invalidity Under 35 U.S.C. § 112, ¶ 1

35 U.S.C. § 112, ¶ 1 requires that the specification contain a written description of the invention. “[T]he hallmark of written description is disclosure.” *Boston Scientific Corp. v. Johnson & Johnson*, 647 F.3d 1353, 1361–62 (Fed. Cir. 2011) (citation omitted). The test for whether a specification adequately describes an invention is “whether the disclosure of the application relied upon reasonably conveys to those skilled in the art that the inventor had possession of the claimed subject matter as of the filing date [T]he test requires an objective inquiry into the four corners of the specification from the perspective of a person of ordinary skill in the art [It] is a question of fact.” *Ariad Pharms., Inc. v. Eli Lilly and Co.*, 598 F.3d 1336, 1351 (Fed. Cir. 2010) (*en banc*); *Boston Scientific*, 647 F.3d at 1362.

The enablement requirement of Section 112 demands that the patent specification enable “those skilled in the art to make and use the full scope of the claimed invention without ‘undue experimentation.’” *Genentech, Inc. v. Novo Nordisk A/S*, 108 F.3d 1361, 1365 (Fed. Cir. 1997)

(quoting *In re Wright*, 999 F.2d 1557, 1561 (Fed. Cir. 1993)). “[T]he scope of the claims must be less than or equal to the scope of the enablement.” *Nat’l Recovery Tech., Inc. v. Magnetic Separation Sys., Inc.*, 166 F.3d 1190, 1196 (Fed. Cir. 1999).

American contends that the following Asserted Claims are invalid under 35 U.S.C. § 112, ¶ 1. Each Asserted Claim identified below (and each Asserted Claim that depends therefrom) is invalid under ¶ 1 because the specification of the Asserted Patent fails to provide a sufficient written description, enabling disclosure, and/or fails to inform, with reasonable certainty, those skilled in the art about the scope of the invention the meaning of the term(s) and/or phrase(s). For each listed term or phrase, American believes the term or phrase is invalid under § 112, ¶ 1, as is any limitation including such terms or phrases, for the same reason(s).

Asserted Patent	Asserted Claim(s)	Term(s) and/or Phrase(s) Lacking Written Description and/or Enablement
'582 patent	1	<ul style="list-style-type: none"> “distributing descriptions of all of said partitions to each of a plurality of subtask processors” “generating at least one output combining all of the subtask outputs and reflecting the processing of all of said subtasks”
'722 patent	14	<ul style="list-style-type: none"> “live object recognizable by the client device” “determine a node type to which the identified category maps” “cohesive respective application environments”
'785 patent	30	<ul style="list-style-type: none"> “virtual network manager” “DNS server for the virtual network ... return a network address associated with a network route director”
'844 patent	7, 11	<p>7:</p> <ul style="list-style-type: none"> “only additional data blocks not previously contained in said root image” <p>11:</p> <ul style="list-style-type: none"> “to create cohesive respective application environments” “at an operational level between file systems”
'469 patent	24, 25, 26, 28, 32	<p>Claim 24:</p> <ul style="list-style-type: none"> “Internet Hotspot” “operatively coupled”

Asserted Patent	Asserted Claim(s)	Term(s) and/or Phrase(s) Lacking Written Description and/or Enablement
		<ul style="list-style-type: none"> • “subscriber access unit” • “authenticating”/“authenticate” • “a remote location experiencing a relatively high volume of transient traffic” • “web-ready device” Claim 25: <ul style="list-style-type: none"> • “the data connection is one of a wired data connection and a wireless data connection” Claim 32: <ul style="list-style-type: none"> • “the wireless connection is one of an 802.11a wireless area network, an 802.11b wireless area network, an 802.11g wireless area network, and an 802.11n wireless area network”
'326 patent	1, 4, 18	Claims 1 & 18: <ul style="list-style-type: none"> • “plurality of data subcarriers” • “partially filling”/ “partially fill” • “one or more guard bands” Claim 4: <ul style="list-style-type: none"> • “transmitting to multiple radios”

2. Invalidity Under 35 U.S.C. § 112, ¶ 2

Claims are indefinite under 35 U.S.C. § 112, ¶ 2 when they “fail to inform, with reasonable certainty, those skilled in the art about the scope of the invention.” *Nautilus, Inc. v. Biosig Instruments, Inc.*, 34 S. Ct. 2120, 2124 (2014). To the extent an asserted apparatus claim includes both apparatus and method limitations, that claim is invalid for indefiniteness under § 112, ¶ 2 because it fails to identify or notify the public of what constitutes direct infringement. *See IPXL Holdings, L.L.C. v. Amazon.com, Inc.*, 430 F.3d 1377 (Fed. Cir. 2005).

In addition to American’s reservation of rights stated above, American will present arguments as to indefiniteness will be presented at that time, i.e., through the Markman meet and confer and briefing process.

American contends that the following Asserted Claims are invalid under 35 U.S.C. § 112, ¶ 2. Each Asserted Claim identified below (and each Asserted Claim that depends therefrom) is

invalid under ¶ 2 because they fail to inform, with reasonable certainty, those skilled in the art about the scope of the claimed invention. For each listed term or phrase, American believes the term or phrase (as well as terms within any such phrase) is invalid under § 112, ¶ 2, and any limitation including such term or phrase is also indefinite.

Asserted Patent	Asserted Claim(s)	Indefinite Term(s) or Phrase(s)
'582 patent	1, 5	<p>1:</p> <ul style="list-style-type: none"> “automatically determining file allocation and logically subdividing records of said input file into a plurality of partitions” “simultaneously executing at least a respective one of the subtasks of the computer-executable process in each of at least some of said processors on a respective one of the partitions with each subtask reading and processing the respective partition so as to process the respective partition and produce respective subtask output” “repeating step (c) in at least some of the subtask processors each with another unprocessed partition on a first-come/first-served basis” <p>5:</p> <ul style="list-style-type: none"> “wherein the output in step (e) is an accumulation of output records from said subtasks in an arbitrary order”
'722 patent	14	<ul style="list-style-type: none"> “live object recognizable by the client device” “gateway device ... configured to identify a category of the update message” “determine a node type to which the identified category maps” “cohesive respective application environments”
'785 patent	30, 37	<p>30:</p> <ul style="list-style-type: none"> “virtual network manager” “receive a registration request from an agent associated with a device” “DNS server for the virtual network ... return a network address associated with a network route director” <p>37:</p> <ul style="list-style-type: none"> “maintain data to associate a virtual network address with a device in the virtual network”

Asserted Patent	Asserted Claim(s)	Indefinite Term(s) or Phrase(s)
'844 patent	7, 11	7: <ul style="list-style-type: none"> “only additional data blocks not previously contained in said root image” “accessed by at least one of said compute nodes” 11: <ul style="list-style-type: none"> “merging the blocks of said root image with the blocks of respective leaf images” “to create cohesive respective application environments” “at an operational level between file systems”
'469 patent	24, 25, 26, 28, 32	Claim 24: <ul style="list-style-type: none"> “Internet Hotspot” “operatively coupled” “subscriber access unit” “authenticating”/“authenticate” “a remote location experiencing a relatively high volume of transient traffic” “web-ready device” Claim 25: <ul style="list-style-type: none"> “the data connection is one of a wired data connection and a wireless data connection” Claim 32: <ul style="list-style-type: none"> “the wireless connection is one of an 802.11a wireless area network, an 802.11b wireless area network, an 802.11g wireless area network, and an 802.11n wireless area network”
'326 patent	1, 4, 18	Claims 1 & 18: <ul style="list-style-type: none"> “full spectral synthesis capability” “combining”/“combine” “plurality of data subcarriers” “partially filling”/ “partially fill” “one or more guard bands” Claim 4 <ul style="list-style-type: none"> “transmitting to multiple radios”

F. Invalidity Under 35 U.S.C. § 101

The following Asserted Claims do not qualify as patent-eligible subject matter and are therefore invalid under 35 U.S.C. § 101.

Asserted Patent	Asserted Claim(s)
'582 patent	1, 2, 5, and 7
'722 patent	14, 16, 17, and 18
'785 patent	30, 35, and 37
'844 patent	7, 11
'469 patent	24, 25, 26, 28, 32
'326 patent	1, 4 ,18

The Court has not yet construed the Challenged Claims. And IV has not yet provided a claim construction for any of the terms or phrases that American anticipates will be in dispute. American, therefore, cannot provide a complete list of § 101 defenses because American does not know whether IV will proffer a construction for certain terms and phrases that is broader than, or inconsistent with, the construction that would be supportable by the disclosure set forth in the specification. American offers these contentions without prejudice to any position they may ultimately take as to any claim construction issue. Accordingly, American reserves the right to supplement, amend, and/or modify these § 101 contentions as discovery progresses.

IV has not yet provided any subject matter eligibility contentions or otherwise disclosed its complete factual and legal basis in opposition to these Preliminary Invalidation Contentions. Further, to the extent related to subject matter eligibility, including describing the industry at the relevant time and each element of each Challenged Claim as found in the prior art, IV also has not provided any validity contentions under 35 U.S.C. §§ 102 or 103. IV also has not provided any expert report on the issue of subject matter eligibility, the inventors have not yet been deposed, and fact and expert discovery is at an early stage. American therefore reserves the right to further amend or supplement these Preliminary Invalidation Contentions in response to any further contentions, factual basis, legal basis, fact discovery, or expert discovery by or of IV asserting that the Challenged Claims are not ineligible under § 101, or that the prior art (as disclosed in these Preliminary Invalidation Contentions and otherwise) does not describe the

industry, at the relevant time, or that any element of each Challenged Claim, both individually and in combination with other elements, was not (i) well understood; (ii) routine; and (iii) conventional. This may include, but is not limited to, identifying additional prior art and other evidence regarding the state of the industry, providing expert testimony, and providing further factual and legal bases under ¶¶ (a)(2)(A) and (a)(2)(B) for why the Challenged Claims are invalid under 35 U.S.C. § 101.

American's § 101 contentions are made in the alternative, and they should not be interpreted, relied upon, or in any way affect the non-infringement or other invalidity arguments American has asserted. American does not infringe any of the Challenged Claims, which are also invalid under 35 U.S.C. §§ 102, 103, and/or 112.

American reserve the right to present expert evidence and testimony in further support of its contention that the Challenged Claims are ineligible under 35 U.S.C. § 101. Such testimony may include evidence and information regarding identifying each exception to eligibility, including why the Challenged Claims are directed to an abstract idea, as well as a description of the industry at the relevant time and what was well understood routine, and conventional in the industry. American specifically reserves the right to present expert testimony on any and all bases, and to do so in its opening expert report on invalidity in the time and manner required by the Court's Scheduling Order.

American incorporates by reference herein its Motion to Dismiss, including its supporting Reply. Additionally, American contends as follows:

1. The '582 patent

The Asserted Claims (claims 1, 2, 5, and 7) of the '582 patent are invalid under 35 U.S.C. § 101 because they are directed to an abstract idea—specifically, the idea of dividing a large task into subtasks and processing them using available resources on a first-come/first-served basis—

and lack any inventive concept that transforms this abstract idea into patent-eligible subject matter. The Asserted Claims of the '582 patent therefore fail at both steps of the Alice test and are therefore invalid. *See e.g. Alice Corp. v. CLS Bank Int'l*, 573 U.S. 208 (2014).

The '582 patent, titled “Method for Managing Distributed Processing of Tasks in a Network,” claims a method for subtask processing in distributed computing systems. Claim 1—the only independent claim of the '582 patent—recites:

1. A method of effecting on a **preexisting input file** a computer-executable process **comprised of a plurality of subtasks**, the method comprising the steps of:

(a) automatically determining file allocation and **logically subdividing records of said input file into a plurality of partitions**;

(b) **distributing descriptions of all of said partitions to each of a plurality of subtask processors**[:];

(c) **simultaneously executing** at least a respective **one of the subtasks** of the computer-executable process in each of at least some of said processors on a respective one of the partitions with each subtask reading and processing the respective partition so as to process the respective partition and produce respective subtask output and;

(d) thereafter **repeating step (c)** in at least some of the subtask processors each **with another unprocessed partition on a first-come/first-served basis**; and

(e) generating at least one output **combining all of the subtask outputs** and reflecting the processing of all of said subtasks.

The patent admits that these steps mostly were well known in the art, including partitioning data, distributing tasks, parallel processing, and aggregating results. '582 patent at 3:35-47, 3:67-4:15, 4:59-63. Partitioning (step a) relies on conventional techniques, such as byte ranges or track addresses, while subtask distribution (step b) employs generic mechanisms using control files; processing subtasks (step c) involves standard read-process-write operations, and aggregation (step e) is “of course” a conventional merging operation. *Id.* at 3:35-4:15, 4:59-63.

The dependent Asserted Claims of the '582 patent fare no better. Claim 2 merely states that a separate processor may perform the partitioning and distribution—an obvious and conventional division of labor that lacks any inventive concept. Claim 5 recites that subtask outputs can be accumulated in arbitrary order but offers no technical detail or mechanism for doing so—just a result. Claim 7 limits the input file to being stored on a network-attached storage device, which the patent itself identifies as a well-known technology.

The claimed first-come/first-served scheduling (step d) was argued during prosecution as the “primary difference” over the prior art. *See* '582 patent, Prosecution History, Mar. 2, 2007 Remarks. According to the applicant, the prior art used “load information to distribute the load between processors,” whereas:

With the instant invention ... load information is not created Instead, the load sharing is done as a byproduct of the fact that the load-sharing process take parts of the load on a first-come/first-served basis. A comparison would be to a road intersection where, according to the prior art, there is a traffic light that determines who can go when. The instant invention is more like such an intersection with a four-way stop so that the individual drivers determine who can go and when.

Id. The patentee’s admission about the “primary difference” between the claimed invention and the prior art is critical in two ways. First, by pinpointing the “first available” assignment technique as the point of novelty, the inventors have defined what the claimed invention is “directed to” for purposes of the *Alice* Step 1 inquiry. *See Affinity Labs of Tex., LLC v. DIRECTV, LLC*, 838 F.3d 1253, 1257–58 (Fed. Cir. 2016) (“The ‘abstract idea’ step of the inquiry calls upon us to look at the ‘focus of the claimed advance over the prior art’ to determine if the claim’s ‘character as a whole’ is directed to excluded subject matter.”). Second, by the simple comparison between the invention and the managing of traffic loads at a street

intersection, the inventor acknowledged that his “invention” merely applies a longstanding method of organizing human activities to computers.

The methods described in the ’582 patent implement that abstract concept using generic computing components, such as processors and memory, which are standard elements in distributed systems. The claims use simple functional language to describe desired outcomes—partitioning, distributing, processing, scheduling, and aggregating—without detailing how these operations are performed or improved. Consequently, the Asserted Claims of the ’582 patent fail at both steps of the Alice test and are therefore invalid.

a) The Asserted Claims of the ’582 patent recite an abstract scheduling technique.

The Asserted Claims of the ’582 patent are directed to the abstract idea of dividing a large task into subtasks and processing them using available resources on a first-come/first-served basis. Claim 1 of the ’582 patent—the only independent claim of the ’582 patent—comprises five steps: (a) “logically subdividing records . . . into a plurality of partitions,” (b) distributing descriptions of those partitions to multiple “subtask processors,” (c) “simultaneously executing” at least one “subtask” on each of the subtask processors, (d) thereafter allocating any remaining “unprocessed partition” to the subtask processors “on a first-come/first-served basis,” and (e) aggregating “all of the subtask outputs” into a final combined output. Stripped of jargon, Claim 1 covers the basic concept of splitting up a task and distributing the work, then collecting the results.

Claim 1’s distributed task management approach is an abstract idea that humans have long practiced without computers; it is akin to breaking a large job into smaller tasks, handing those tasks out to a group of workers, giving each new work as they free up, and then assembling the final product. The practice of breaking up a large job into discrete tasks, passing out a first

round of assignments to team members, instructing them to ask for a new assignment when finished, and then combining the results has been employed by humans since the building of the Egyptian pyramids. The Federal Circuit has consistently held that processes based on such fundamental organizational practices are abstract. *See, e.g., Intellectual Ventures I LLC v. Capital One Bank (USA)*, 792 F.3d 1363, 1367 (Fed. Cir. 2015) (organizing, storing, and retrieving information held abstract); *Intellectual Ventures I LLC v. Symantec Corp.*, 838 F.3d 1307 (Fed. Cir. 2016) (automatically organizing information into wanted and unwanted categories and discarding the unwanted information held abstract).

“[T]here is a critical difference between patenting a particular concrete solution to a problem and attempting to patent the abstract idea of a solution to the problem in general.” *Electric Power Group, LLC v. Alstom S.A.*, 830 F.3d 1350, 1356 (Fed. Cir. 2016). The Asserted Claims of the ’582 patent fall on the latter side of that line: they seek to monopolize the general idea of a solution (parallel task processing with on-the-fly scheduling) rather than a concrete technological solution. The Asserted Claims of the ’582 patent do not improve computer technology or solve a specific technical problem in how computers operate but instead apply conventional and common-sense workload management ideas to distributed computing. Partitioning data, distributing tasks, processing in parallel, and aggregating results are standard operations in distributed computing systems. ’582 patent at 3:35–4:15, 4:59–63. Simply executing a known business/workflow practice on generic computers is not a patentable improvement to those computers.

The ’582 patent’s recitation of assigning work using first-come/first-served scheduling—a ubiquitous method of task processing akin to queue management (e.g., a line at a concession stand or a four-way stop)—demonstrates its abstract nature. Claim 1 applies this principle to

parallel computer processing without offering any new technological innovation. Moreover, The Asserted Claims of the '582 patent use abstract functional language to describe desired outcomes—such as “logically subdividing,” “distributing,” and “aggregating”—without detailing how these steps are performed. This kind of result-oriented claiming that specifies an outcome (efficient parallel processing) without limiting the claim to any innovative technique for achieving it is a hallmark of an abstract idea. Thus, the '582 patent “merely applies a well-known idea using generic computers ‘to the particular technological environment of the Internet.’” *Intellectual Ventures I LLC v. Symantec Corp.*, 838 F.3d 1307, 1314 (Fed. Cir. 2016). In *Two-Way Media*, the Federal Circuit held that reciting a series of steps to achieve a desired result, without specifying how those steps are performed, renders the claim abstract. *See Two-Way Media Ltd. v. Comcast Cable Commc'ns, LLC*, 874 F.3d 1329, 1337–38 (Fed. Cir. 2017) (“routing” and “accumulating records” held abstract and invalid). Even though the specification describes conventional methods for these operations, the Asserted Claims of the '582 patent do not specify how data is partitioned, how partitioned subtasks are distributed, or how results are aggregated. '582 patent at 3:35–4:15, 4:59–63. Such high-level, results-oriented language underscores the claim’s abstract nature. As the Federal Circuit noted in *Electric Power*, claims that focus on the outcome of data processing without specifying the means to achieve it are abstract and invalid. 830 F.3d at 1356.

The '582 patent are not directed to an improvement in data processing because using multiple computers in parallel may increase the speed of execution of processes and balances the load on resources. Dkt. 28 at 9-10. To the extent that the claimed method achieves these benefits, it only does so by applying the abstract idea of task partitioning and parallel execution on generic computers, rather than by reciting a new technology. Improved speed and load balancing are

expected results of running a workload in parallel; they are performance outcomes, not inventive technical means. The '582 patent claims do not improve the internal operation of the computer processors or change how the network functions—they just use more computing power in a straightforward way. Merely accelerating an existing process via parallelization is akin to the scenario in *SAP America v. InvestPic*, where claims “selecting certain information, analyzing it using mathematical techniques, and reporting or displaying the results of the analysis” were held abstract. 898 F.3d 1161, 1167-1168 (Fed. Cir. 2018). Indeed, the claims of the '582 patent “fit into the familiar class of claims that do not ‘focus ... on [] an improvement in computers as tools, but on certain independently abstract ideas that use computers as tools.’” *Id.* at 1168 (quoting *Electric Power*, 830 F.3d at 1354). Furthermore, IV’s reliance on *Enfish, LLC v. Microsoft Corp.*, 822 F.3d 1327 (Fed. Cir. 2016), is misplaced. Unlike *Enfish*, where the claims improved database memory structure, the '582 patent does not specify any new hardware or improvement to system architecture—it merely applies a standard method of parallelization to a computing process.

Because the Asserted Claims of the '582 patent of the '582 patent are directed to a fundamental method for organizing and processing tasks, it falls squarely within the category of claims deemed ineligible under Step 1 of the *Alice* framework.

b) The '582 patent lacks an inventive concept.

As to *Alice* Step 2, The Asserted Claims of the '582 patent recite only conventional steps and generic components without introducing any meaningful technical improvements or innovations. The claim elements, individually and in combination, reflect routine operations in distributed computing, as explicitly acknowledged in the specification. Partitioning input data into subsets, distributing tasks to processors, and processing tasks in parallel were standard techniques for optimizing workload distribution at the time. '582 patent at 3:35-4:15, 4:59-63.

The first-come/first-served scheduling technique, likewise, is a well-established method for processing tasks based on availability, like the applicant’s example of the four-way stop sign. The Asserted Claims of the ’582 patent “merely appl[y] [this] well-known idea using generic computers.” *Symantec*, 838 F.3d at 1314. The specification does not describe any modification, improvement, or adaptation of this scheduling technique to make it unique or unconventional in the context of distributed computing. The Asserted Claims of the ’582 patent simply apply this known concept predictably to allocate unprocessed subtasks, a principle embedded in all multi-threaded and distributed computing architectures and offers no specific mechanisms or technical enhancements. Thus, the Asserted Claims of the ’582 patent are “directed to a result or effect that itself is the abstract idea and merely invoke[s] generic processes and machinery” rather than “a specific means or method that improves the relevant technology.” *Yu v. Apple Inc.*, 1 F.4th 1040, 1043 (Fed. Cir. 2021). “What is claimed is simply a generic environment in which to carry out the abstract idea.” *Id.*

Confirming the routine nature of the claimed methods, both the specification and the Asserted Claims of the ’582 patent rely entirely on generic computing components, such as processors, memory, and standard network architecture. ’582 patent at 4:27–36; *Trinity Info Media, LLC v. Covalent, Inc.*, 72 F.4th 1355, 1364, 1366–67 (Fed. Cir. 2023) (generic computer components such as “processors” and “memory” insufficient to add an inventive concept). The patent does not propose any novel configurations or functionalities for these components but instead recites conventional “functions in general terms, without limiting them to technical means for performing the functions that are arguably an advance over conventional computer and network technology.” *Elec. Power*, 830 F.3d at 1351.

For example, Claim 1’s reliance on functional language—such as “automatically determining,” “logically subdividing,” “distributing,” “executing,” and “combining”—describes desired outcomes without specifying how those outcomes are achieved. It does not explain how data is partitioned into subsets, how subtasks are allocated, how unallocated tasks are managed, or how results are aggregated into a final output. ’582 patent at 6:47–4:15, 4:59–63. This lack of specificity leaves the claim framed at a high level of abstraction, with no concrete implementation details that could transform the method into a patent-eligible application. The dependent Asserted Claims of the ’582 patent fare no better. Claim 2 merely delegates partitioning and distribution to another processor, without explaining how that processor performs the task. Claim 5 generically allows output in arbitrary order but lacks any technical detail on how that’s achieved. Claim 7 limits the input file to network-attached storage—a conventional environment the patent admits was already well known.

Courts have consistently held analogous claims ineligible under Step 2 of the Alice framework. In *Content Extraction*, 776 F.3d at 1348, the court found “no ‘inventive concept’” in the use of generic computer components to perform routine activities. Similarly, in *Symantec Corp.*, 838 F.3d at 1320, the Federal Circuit emphasized that for a computer-implemented invention to be patent-eligible, it must involve more than well-understood, routine, and conventional activities. Like the claims in these cases, the Asserted Claims of the ’582 patent merely apply abstract principles of task management to distributed computing using conventional techniques, which does not suffice to render it patent-eligible.

The patent’s description of using heterogeneous computers and balancing load is not evidence that the claims capture a technical innovation. As detailed above, the combination of steps in the Asserted Claims of the ’582 patent are entirely conventional—they are the standard

recipe for parallel processing (split, distribute, process, repeat, merge). There is no inventive coordination or unconventional synergy arising from the arrangement of these steps; each step does what one would normally expect, and together they achieve the expected outcome of parallel execution. The Asserted Claims of the '582 patent simply assemble known techniques to carry out an abstract goal. Combining conventional steps in a computer environment, without more, does not create an inventive concept. IV emphasizes that the patent describes using multiple heterogeneous computers and optimizing execution, but those aspects were already familiar in distributed computing.

Viewed as a whole, the Asserted Claims of the '582 patent combine conventional steps in a predictable manner to achieve the abstract goal of managing distributed tasks. They offer no technological advancements or inventive concepts beyond the abstract idea itself. Accordingly, the '582 patent fails to meet the requirements for patent eligibility under 35 U.S.C. § 101.

2. The '722 patent

The Asserted Claims (claims 14, 16, 17, and 18) of the '722 patent recite the abstract idea of receiving information, identifying relevant content, registering interest, and disseminating updates to interested parties. This is a long-standing and fundamental practice in organizing and distributing information—not a technological improvement or specific technical solution. Because the claims implement an abstract idea using only generic networking and data-handling functions without reciting any technological improvement or unconventional hardware, the claims fail both steps of the *Alice* framework.

a) The Asserted Claims of the '722 patent are Directed to an Abstract Idea

The Asserted Claims of the '722 patent are directed to the abstract idea of receiving information, identifying relevant content, registering interest, and disseminating updates to

interested parties. Claim 14—the only representative independent claim—describes a method for enabling dynamic updates to “live objects” displayed on a client device through a structured communication framework involving a source device, a routing network, and intermediary nodes. An input source sends a data representation to a client device via a routing network; this data includes at least one live object—an identifiable element capable of receiving updates. The client device recognizes the live object, determines its identifier, and registers with the routing network to receive updates about it. This registration includes connection information, which is stored at a node in the routing network. When the input source sends an update message for the live object, the message is routed through a gateway that determines its category and selects the appropriate node type to handle it. The node then identifies the registered client and forwards the message. Upon receipt, the client updates the relevant property of the live object.

Claim 16 adds to claim 14 that the client stores metadata related to the live object, and that the update message modifies this metadata. Claim 17 adds to claim 14 that the update message includes a message type, which the client uses to determine how to apply the update. Claim 18 expands on claim 14 by allowing the client to register for multiple live objects and receive updates for each.

What the Asserted Claims of the '722 patent effectively describe is a rules-based information forwarding mechanism, based on identifiers and interest registration—a framework that has existed in mailrooms, subscription services, and newsrooms long before the advent of digital networks. Its implementation using computers and a “routing network” merely automates this longstanding human practice.

Stripped of jargon, the Asserted Claims of the '722 patent describe a familiar, abstract pattern: sending information, having a recipient identify what is relevant, and distributing future

updates to those who have expressed interest. This pattern is not new or technological. It is, for example, identical in substance to how a traditional newsroom operates. A journalist (the “input source”) provides a summary of several developing stories (the “data representation”) to a central bulletin board in the newsroom. Each story on the board is a “live object” in the sense that it may be updated. Editors (the “client devices”) review the bulletin board and mark their interest in particular stories. When updates come in from the field, a receptionist (the “gateway device”) categorizes the update and determines which editor (the “node”) should receive it. The update is then delivered to the relevant editor, who revises the draft story accordingly.

This real-world newsroom example illustrates that the Asserted Claims of the ’722 patent merely recite an abstract method of organizing and delivering information—a practice that has been implemented manually for decades. The fact that the ’722 patent describes this process in the context of a computing environment does not render it any less abstract. Claims directed to abstract frameworks for information delivery—like updating content in response to user interaction or preference—have repeatedly been held patent-ineligible. *See, e.g., Broadband iTV, Inc. v. Amazon.com, Inc.*, No. 6:20-CV-00921-ADA, 2022 WL 4703425, at *11 (W.D. Tex. Sept. 30, 2022) (J. Albright), *aff’d*, 113 F.4th 1359 (Fed. Cir. 2024); *see also Broadband iTV, Inc. v. Oceanic Time Warner Cable, LLC*, 135 F. Supp. 3d 1175, 1178 (D. Haw. 2015) (claims directed to abstract idea of “using the same hierarchical ordering based on metadata to facilitate the display and locating of video content” ineligible under § 101). Because the Asserted Claims of the ’722 patent focus on the result of an implementing an abstract idea—delivering updated content to interested recipients—rather than a specific technological means of achieving that result, they fail at *Alice* step one.

The specification attempts to present these steps as an improvement over prior art by claiming to enhance network efficiency through dynamic client registration and message routing. It asserts that the use of node specialization and dynamic mapping reduces unnecessary message traffic. The routing network purportedly eliminates the need for the input source to track which clients are displaying the live objects by maintaining a registry of registered clients. The system is described as being useful for applications where content on client devices must be updated dynamically based on changes in data, such as stock prices or sports scores. However, the specification itself reveals that the claimed methods are implemented using conventional networking techniques. The routing network, which consists of nodes and gateways, is built using generic computing components, including single-processor computer systems running common operating systems like Linux. The specification indicates that the system utilizes existing messaging protocols, such as HTTP, and known client-server architectures. Furthermore, the patent acknowledges that the basic concepts of client registration and update handling were already present in existing methods that included client-driven approaches, applets, and server-driven updates using TCP/IP connections.

b) The Asserted Claims of the '722 Patent Do Not Recite an Inventive Concept

Step 2 of *Alice* cannot save the Asserted Claims of the '722 patent from ineligibility under 35 U.S.C. § 101. The Asserted Claims of the '722 patent do no more than implement the abstract idea of selectively distributing content updates to interested recipients using generic computing components performing routine, conventional functions. The purported improvement—routing update messages based on client registration—does not introduce any new computing technology or method of enhancing network efficiency. Instead, it merely applies well-known client-server communication techniques to the problem of distributing data updates.

The specification itself discloses that the system utilizes conventional networking components and known data transmission protocols without any significant modification.

Claim 14 recites basic components: an “input source,” a “client device,” a “routing network,” “nodes,” and “gateways.” But none of these are claimed in a specific or inventive configuration. The specification describes the input source as a web server or dynamic content provider, the client device as a browser-enabled device such as a PC or mobile phone, and the routing network as an internet-based message delivery structure. The “gateway” and “node” elements, while given roles in the message flow, are described in general terms that mirror conventional content delivery mechanisms—such as proxies, relays, load balancers, or intermediate servers.

There is no technical detail in the claim about how messages are classified, how registration is implemented, or how update routing is performed. The claim merely recites the expected result: that a client receives an update for a live object it previously registered for. Courts have routinely held that such claims, which rely on routine network behavior and leave implementation to the practitioner, do not supply an inventive concept. *See, e.g., Two-Way Media Ltd. v. Comcast Cable Commc’ns, LLC*, 874 F.3d 1329, 1339 (Fed. Cir. 2017) (“The claim uses a conventional ordering of steps—first processing the data, then routing it, controlling it, and monitoring its reception—with conventional technology to achieve its desired result.”).

The specification itself confirms that the invention is designed to operate on existing infrastructure and leverages known technologies. Client devices are said to use standard web browsers with existing support for Java, JavaScript, or plug-ins. Update messages are transmitted over HTTP and are parsed using client-side scripts. The routing network is implemented using conventional clusters, nodes, and gateways, and messages are routed using category/type

mappings or registry lookups—well-understood design patterns in networked systems. The so-called “live objects” are simply HTML elements or other page components that can be identified by an ID and updated via DOM manipulation or script execution, which common techniques in web development since at least the late 1990s. The use of an “activation module” to register interest in updates is just another name for client-side code that subscribes to a push service—an architecture long practiced in content syndication and notification systems. Nothing in Claim 14 requires, or even enables, a novel technical implementation of these mechanisms.

None of the limitations in claims 16, 17, or 18 inject any inventive concept into the ’722 patent:

- **Claim 16**’s mention of metadata storage and update is a routine feature of client-side state tracking. Web applications and software clients have long-maintained associated metadata (e.g., timestamps, tags, formatting) alongside core content.
- **Claim 17** introduces a message type field used to determine how to process an update. But conditional logic based on message types is a staple of basic programming and messaging protocols. This is no more inventive than a “switch” statement tied to a message header.
- **Claim 18** expands registration to multiple live objects, which is a straightforward extension of the core idea. There is no novel mechanism for batch registration, grouping, or prioritization; the claim merely allows more than one subscription, a feature inherent in any scalable publish-subscribe system.

Thus, the dependent claims likewise apply well-known programming constructs and user-experience features without claiming how any of them are implemented in a novel or unconventional manner.

Furthermore, the combination of client registration and dynamic message routing was not an inventive concept at the time of the alleged invention. Technologies such as IP Multicast, Java Message Service (JMS), and CORBA Event Services were already implementing message routing based on client registration well before the priority date. The combination of these known elements does not produce unexpected results or provide a technological improvement. Here, the claimed method merely takes existing client registration and message routing techniques and applies them in a straightforward, predictable way.

The Asserted Claims of the '722 patent therefore do not improve any specific computer functionality or solve any technical problem in a new way—they merely implement a conventional information delivery model using abstract functional language and general-purpose computing components. As such, the Asserted Claims of the '722 patent fail *Alice* step two and are invalid under 35 U.S.C. § 101.

* * *

The Asserted Claims of the '722 patent are invalid under 35 U.S.C. § 101 because they fail both prongs of the *Alice* framework. At step one, these claims are directed to the abstract idea of distributing updates about selected content to interested recipients—an organizing principle that has long been practiced by humans and implemented in conventional information systems. These claims describe this idea at a high level of generality, using functional language untethered to any specific improvement in technology or network design. At step two, the claims do not recite any inventive concept sufficient to transform this abstract idea into a patent-eligible application. The components are generic, the operations are conventional, and the implementation relies entirely on known computing techniques. These claims simply automate a

familiar content delivery paradigm using routine architecture, offering no technical advancement over the prior art.

3. The '844 patent

The Asserted Claims (claims 7, 11) of the '844 patent recite the abstract idea of organizing and storing information using a “root-leaf” structure. This is a long-standing and fundamental practice in organizing and distributing information—not a technological improvement or specific technical solution. Because the claims implement an abstract idea using only generic networking and data-handling functions without reciting any technological improvement or unconventional hardware, the claims fail both steps of the *Alice* framework.

a) The Asserted Claims of the '844 Patent are Directed to an Abstract Idea

The Asserted Claims of the '844 patent are directed to the abstract idea of organizing and storing information using a “root-leaf” structure. Although applied to a computer network, the method steps of independent claim 7—storing root and leaf images and caching accessed blocks—mirror basic data collection and organization practices, analogous to longstanding human activities. The patent attempts to claim the concept of storing some information in a central repository for permanent storage and common use (a root image), storing other non-identical data locally for faster retrieval and personal use (leaf images), and keeping recently used items temporarily on hand (cached). Such abstract ideas for how to organize and store data on computer systems are one of “the ‘basic tools of scientific and technological work’ that are free to all men and reserved exclusively to none.” *Synopsys, Inc. v. Mentor Graphics Corp.*, 839 F.3d 1138, 1146 (Fed. Cir. 2016) (quoting *Alice*, 573 U.S. at 216). Courts have consistently held such practices abstract. *See Intellectual Ventures I LLC v. Erie Indem. Co.*, 850 F.3d 1315, 1327 (Fed. Cir. 2017) (organizing and accessing data held abstract); *Content Extraction &*

Transmission LLC v. Wells Fargo Bank, Nat'l Ass'n, 776 F.3d 1343, 1347 (Fed. Cir. 2014) (data collection and storage held abstract); *Kaavo Inc. v. Amazon.com Inc.*, 323 F. Supp. 3d 630, 641 (D. Del. 2018) (“setting up and managing a cloud computing environment” held abstract); *Versata Software, Inc. v. NetBrain Techs., Inc.*, No. 13-676-LPS-CJB, 2015 WL 5768938, at *7 (D. Del. Sept. 30, 2015) (“representing information in a hierarchy amounts to an abstract idea.”).

The specification confirms that the described components and techniques, including root-leaf storage systems and caching mechanisms, were well understood at the time of the invention. Dkt. 1-1 at 1:31–45, 6:37–7:5. The Asserted Claims of the '844 patent provide no specifics on storage structures/hardware, caching algorithms/techniques, or other technical improvements, relying instead on functional language that describes high-level results without specific implementation details. *See ChargePoint, Inc. v. SemaConnect, Inc.*, 920 F.3d 759, 769 (Fed. Cir. 2019) (“Even a specification full of technical details about a physical invention may nonetheless conclude with claims that claim nothing more than the broad law or abstract idea underlying the claims.”).

Claim 7's scant details about what data gets saved where are inherently abstract. *See Content Extraction*, 776 F.3d at 1347 (“The concept of data collection, recognition, and storage is undisputedly well-known. Indeed, humans have always performed these functions.”). “This type of activity, *i.e.*, organizing and accessing [data], includes longstanding conduct that existed well before the advent of computers and the Internet.” *Erie Indem.*, 850 F.3d at 1327. Claim 7 includes no purportedly novel technical details for cluster computing—just an abstract idea for how to organize and store data. In the end, the claim is little different from a librarian storing different types of books and papers in different sections of a library. Claim 7 therefore fails *Alice* Step 1 because it recites a conventional root-leaf storage system using a high-level abstract

concept to organize and store data without any specific technological improvement recited in the claim. Claim 11 fares no better; it merely adds a generic “merging” step—joining the blocks of the root and leaf images to “create cohesive respective application environments”—without describing *how* that merging is performed or any technical rules guiding it. Claim 11 therefore provides no new or inventive mechanism for merging, just the result.

b) The Asserted Claims of the '844 Patent Do Not Recite an Inventive Concept

The Asserted Claims of the '844 patent likewise fail Step 2, in part, because “these claims use generic computers to perform generic computer functions.” *Symantec*, 838 F.3d at 1315. The specification concedes that the claimed methods are standard, including storing root images on a first storage unit, leaf images on second storage units, and caching accessed blocks. Dkt. 1-1 at 2:14–21, 6:38–67. Thus, even if the specification discloses ideas that could be applied to useful ends, claims 7 and 11 “contain[] no restriction on how the result is accomplished.” *Symantec*, at 1316.

The claimed steps are described at a high level of generality, with no new algorithms, configurations, or technological improvements. Generic terms like “providing,” “storing,” and “caching” describe desired outcomes without specifying any “inventive concept” for how these tasks are performed. *See BSG Tech*, 899 F.3d at 1290-91 (“If a claim’s only ‘inventive concept’ is the application of an abstract idea using conventional and well-understood techniques, the claim has not been transformed into a patent-eligible application of an abstract idea”); *Elec. Power Grp., LLC v. Alstom S.A.*, 830 F.3d 1350, 1356 (Fed. Cir. 2016) (drawing the “distinction between ends sought and particular means of achieving them, between desired results (functions) and particular ways of achieving (performing) them”).

Breaking down each element of the Asserted Claims of the '844 patent reveals how each element is conventional:

Claim 7:

- **“storing blocks of a root image of said compute nodes on a first storage unit”**

This is conventional. Shared read-only root images were routinely stored on central storage devices, as used in Sun’s diskless workstations and LTSP environments. (*See Hitz et al.; NetApp WAFL*).

- **“storing leaf images for respective compute nodes on respective second storage units...”** This element is likewise conventional. Systems such as

VMware and Episode stored deltas or overlays for individual machines using CoW. These deltas excluded unchanged base data, which was referenced from the shared root image.

- **“caching blocks of said root image that have been accessed...”**

This element is likewise conventional. Caching mechanisms, including block-level caching and LRU algorithms, were widely used by 2004 in both operating systems and storage appliances (*see NetApp; LTSP*).

Claim 11:

- **“merging said blocks of the root image and said leaf images...”**

The element recited by Claim 11 is similarly conventional and routine. The merging recited by Claim 11 had been widely implemented in systems such as VMware’s and NetApp’s platforms.

Courts have invalidated analogous claims. In *Symantec Corp.*, 838 F.3d 1307, the Federal Circuit invalidated claims for managing email messages using conventional computer

components. Similarly, the court in *Content Extraction*, 776 F.3d 1343, held claims for data recognition and storage invalid because they relied on routine computing methods. The Asserted Claims of the '844 patent fall into the same category, as they merely recite routine computing techniques and hardware.

The Asserted Claims of the '844 patent offer no inventive step, combining conventional techniques predictably to achieve abstract goals. They rely on generic computing elements and routine data management practices, failing *Alice* Step 2. Thus, The Asserted Claims of the '844 patent are ineligible under §101.

* * *

Nonetheless, at their core, claims 7 and 11 are directed to the abstract idea of organizing and storing data using a hierarchical structure of a shared base and individualized overlays. The '844 patent relies entirely on general-purpose computing devices with generic storage media (Dkt. 1-1 at 4:27–56), and unclaimed components, such as “Union Block Devices” (UBDs), described as low-level drivers with no disclosed novel functionality (*id.* at 5:19–64). The Asserted Claims of the '844 patent likewise employ broad functional language like “providing,” “storing,” and “caching,” without specifying any improved algorithms, hardware, or software. Thus, the Asserted Claims of the '844 patent merely recite an abstract idea for organizing and storing data without any inventive concept, thereby failing at both steps of the *Alice* test.

4. The '785 patent

The Asserted Claims (claims 30, 35, and 37) of the '785 patent recite the abstract idea of membership management and address look-up in a communications network. This is a long-standing and fundamental practice in administrative tasks—not a technological improvement or specific technical solution. Because the claims implement an abstract idea using only generic

administrative functions without reciting any technological improvement or unconventional hardware, the claims fail both steps of the *Alice* framework.

a) The Asserted Claims of the '785 Patent are Directed to an Abstract Idea

The Asserted Claims of the '785 patent are directed to the abstract idea of membership management and address look-up in a communications network—mere administrative tasks performed every day by receptionists, security personnel, and bouncers. Stripped of jargon, the claimed elements in independent claim 30 are basic administrative functions involved in setting up and maintaining a private communication network—operations that map directly onto longstanding abstract concepts such as maintaining a directory of users, assigning identifiers, and responding to queries with directing information. Claim 35 builds on claim 30 by adding a “join module” that receives join and leave requests—analogueous to a sign-in/sign-out log for network members. Claim 37 further adds that the join module maintains a mapping between virtual addresses and devices—a conventional association table akin to a user directory or ARP (address resolution protocol) table.

The Asserted Claims of the '785 patent all reflect the abstract idea of membership management and address look-up in a communications network—akin to a corporate office receptionist managing an internal directory. The receptionist receives visitors (devices) requesting access (registration), checks whether they're authorized to enter (authentication), assigns them a visitor badge with an internal extension number (virtual address), and, when asked, tells other employees where to find the visitor (DNS resolution and routing). The receptionist also tracks when the visitor checks in and out (join and leave requests) and keeps a temporary log mapping badge numbers to visitors (address-device association). These are routine administrative functions that merely facilitate communication within a defined space—just as the

Asserted Claims do within a virtual network. There is no innovation in how the receptionist performs these tasks, and these tasks have been performed by humans for centuries. Likewise, the Asserted Claims do not improve any underlying networking technology or propose any inventive technical implementation. The Asserted Claims of the '785 patent are simply automating a familiar organizational role using conventional computing tools, rendering them abstract under *Alice* step one.

Tellingly, the method of originally filed claim 64 (which issued as claim 48) did not include any requirement that the steps of the claim be performed by a computer. On February 4, 2009 (5 years before the *Alice* decision), the Examiner rejected claim 64 under 35 U.S.C. § 101, applying the *Bilski* “machine-or-transformation” test for patentable subject matter that existed in 2009. In other words, without the recitation of a “machine” (*i.e.*, a computer), the claims were not patentable under § 101. In response to the Section 101 rejection, the Applicants amended claim 64 to be “computer-implemented,” as none of the originally-filed claims required the use of a computer. Although an amendment requiring that the method be performed by a computer was sufficient in 2009-2010 to overcome the Examiner’s Section 101 rejections, this is not the case today. Under current Supreme Court and Federal Circuit law, the *Alice* steps apply.

The '785 patent applicants’ computer-implemented amendment to independent claim 64 crystallizes two points. First, the amendment demonstrates that the Applicants understood that the method described in the '785 patent and claimed in the originally-filed claim 64 could be practiced without the use of a computer, *i.e.*, the method could be performed in the human mind, or by a human using a pen and paper. *See CyberSource Corp. v. Retail Decisions, Inc.*, 654 F.3d 1366, 1372–73 (Fed. Cir. 2011) (holding that a claim whose “steps can be performed in the human mind, or by a human using a pen and paper” is directed to an “unpatentable mental

process[.]”); *Elec. Power Grp.*, 830 F.3d at 1355 (“[C]laims focusing on organizing, characterizing, and/or sorting information are patent-ineligible because they are not significantly differentiable from ordinary mental processes.”). Second, this amendment demonstrates that the focus of the claims of the ’785 patent is not on any specific asserted improvement in computer capabilities. *In re TLI Commc’ns L.L.C. Pat. Litig.*, 823 F.3d at 612 (“[T]he claims were not directed to specific improvement to computer functionality. Rather, they are directed to the use of conventional or generic technology in a nascent but well-known environment, without any claim that the invention reflects an inventive solution to any problem presented by combining the two.”).

Under *Alice*, a claim that is unpatentable without the recitation of a computer does not become patentable by reciting a computer. That the claims of the ’785 patent merely implement the non-statutory subject matter on a generic computing device is plain from the fact that the originally-filed claims did not require that the claimed systems and methods include or be performed by a computer, respectively. Thus, the language of the Asserted Claims, their prosecution history, and the change in the law demonstrate that the Asserted Claims of the ’785 patent are directed to an abstract administrative idea applied to a communications network.

b) The Asserted Claims of the ’785 Patent Do Not Recite an Inventive Concept

The Asserted Claims of the ’785 patent fail at *Alice* step 2 because they do not recite anything beyond generic computer functions performed using conventional components. The specification confirms that the system is built using well-known networking elements (*e.g.*, processors, memory, interfaces, DNS servers, NAT traversal techniques, virtual IP addressing) without any specialized circuitry, protocol, or configuration.

The '785 patent itself acknowledges that the components and methods it employs were conventional at the time of invention. For example, the specification discusses the use of Network Address Translation (NAT) and virtual addresses to facilitate secure communication, but these techniques were already well-known in the field of networking. The patent does not propose any novel way of utilizing NAT or managing virtual addresses beyond their ordinary and conventional uses.

Claim 30, for instance, merely describes using a DNS server to resolve device addresses within a virtual network, which is a standard function of DNS servers generally. The idea of assigning virtual addresses to devices and resolving these addresses through DNS lookups is not unique or inventive. Instead, the patent recites high-level results—such as secure communication—without specifying how these results are technologically achieved. There is no disclosed protocol, no new addressing scheme, no algorithm for conflict resolution, and no structural innovation in how devices are registered or managed. The system instead relies on existing IP, UDP, NAT, and IPsec technologies. The claims are results-oriented: “register the device,” “assign an address,” “respond to a DNS query”—without any accompanying inventive technical solution.

The claim elements—processor, memory, network interface, modules—are standard hardware or logical software components performing their expected functions. There is no claim that the invention requires new hardware, reconfigured components, or even custom protocols. The Asserted Claims of the '785 patent are therefore equivalent to the claims in *BSG Tech LLC v. BuySeasons, Inc.*, 899 F.3d 1281, 1291 (Fed. Cir. 2018), where the claims at issue merely limited an abstract idea to a particular technological environment and were held ineligible. Similarly, in *ChargePoint, Inc. v. SemaConnect, Inc.*, the court invalidated claims that combined

generic networking capabilities with existing concepts without adding any inventive technology. 920 F.3d 759, 774 (Fed. Cir. 2019) (“But network control is the abstract idea itself, and ‘a claimed invention’s use of the ineligible concept to which it is directed cannot supply the inventive concept that renders the invention ‘significantly more’ than that ineligible concept.’”) (quoting *BSG Tech*, 899 F.3d at 1290). The Asserted Claims of the ’785 patent follow the same pattern by applying well-understood networking techniques without advancing the state of the art.

* * *

The Asserted Claims of the ’785 patent are invalid under 35 U.S.C. § 101. They are directed to abstract administrative operations for managing virtual networks and fail to recite any inventive concept in either their individual components or overall arrangement. They do not improve any underlying technology, nor do they disclose a novel technical mechanism for achieving the stated goals. As such, they fail both steps of the *Alice* test and are ineligible for patent protection.

5. The ’326 patent

a) The Asserted Claims of the ’326 Patent are Directed to an Abstract Idea

The Asserted Claims of the ’326 patent recite the abstract idea of combining channels to increase data throughput (claims 1 and 18) and transmitting data to multiple radios (claim 4). These claims fail Step 1 of *Alice*. See *Alice*, 573 U.S. at 216-17. They recite long-standing and fundamental practices in sending and receiving information over a network—not a technological improvement or specific technical solution. There is nothing to the patent beyond the concept of combining two data transmission channels together, including the space between them, to yield a larger channel.

Channel bonding and transmitting data to multiple radios is an abstract idea as claims directed to sending and receiving data are abstract in nature. *See, e.g., Affinity Labs of Texas, LLC v. DIRECTTV, LLC*, 838 F.3d 1253, 1261 (Fed. Cir. 2016) (explaining that patents directed to “the conveyance and manipulation of information using wireless communication and computer technology” do not disclose eligible subject matter). Claims directed to formatting data are also abstract. *See Dropbox, Inc. v. Synchronoss Techs., Inc.*, 815 F. App’x 529, 537 (Fed. Cir. 2020) (finding claims ineligible that were directed, in part, to “formatting data”). Additionally, the Asserted Claims of the ’326 patent are directed towards manipulating data subcarriers, filling frequency gaps, and operations such as Fast Fourier Transforms, all of which are fundamentally mathematical operations. The Asserted Claims are thus invalid, as “[p]urely conventional or obvious pre-solution activity is normally not sufficient to transform an unpatentable law of nature into a patent-eligible application of such a law.” *Mayo Collaborative Services v. Prometheus Labs., Inc.*, 566 U.S. 66, 79, (2012) (cleaned up); *see also Digitech Image Techs., LLC v. Elecs. for Imaging, Inc.*, 758 F.3d 1344, 1351 (Fed. Cir. 2014) (“Without additional limitations, a process that employs mathematical algorithms to manipulate existing information to generate additional information is not patent eligible.”).

b) The ’326 Patent Claims Lack an Inventive Concept.

Step 2 of *Alice* does save the Asserted Claims of the ’326 patent from ineligibility under 35 U.S.C. § 101. The Asserted Claims of the ’326 patent do no more than implement the abstract idea of increasing data throughput by using generic computing components performing routine, conventional functions. Namely, channel bonding (claims 1 and 18) and transmitting data to multiple radios (claim 4) lack an inventive concept because such claims amount only to generic components and fail to demonstrate unconventionality. *See Dropbox*, 815 F. App’x at 537

(finding that “data transfer connection[s] are generic computer-related concepts as they literally are just data transfer modalities” and are thus not inventive concepts).

The Asserted Claims of the ’326 patent are invalid under 35 U.S.C. § 101. They are directed to the abstract idea of transferring data over a network and fail to recite an inventive concept. Because the claims implement an abstract idea using only generic networking and data-handling functions without reciting any technological improvement or unconventional mechanisms, the claims fail both steps of the Alice framework.

6. ’469 patent

a) The Asserted Claims of the ’469 Patent recite the business practice of placing a paid hotspot in a “transient location”

Each of the Asserted Claims of the ’469 patent are invalid under 35 U.S.C. § 101 because the claims are directed to an abstract idea and fail to describe an inventive concept. *Alice Corp. v. CLS Bank Int’l*, 573 U.S. 208, 216-17 (2014). For example, all of the Asserted Claims of the ’469 patent are generally directed to an abstract idea for Section 101 purposes and under Intellectual Venture’s apparent constructions. The sole asserted independent claim, claim 24, concerns placing a known composition of commercial components in a “remote location” to establish a subscribers-only internet hotspot. The specification also makes clear that the invention of the Asserted patent is the idea of placing paid hotspots where users would otherwise not have Internet access:

The present invention provides rural “Hotspots” (such as Wi-Fi access, for example) to enable wireless and hardwired, satellite distributed Internet access for anyone with a PC or other web-ready device (wireless ready or cabled) and a valid credit card.

’469 patent at 1:34-38. The concept of charging people money for internet access in a “transient location” like a restaurant or motel is merely a business idea. Paying for services is a “fundamental economic practice long prevalent in our system of commerce” and, “as such, is a

patent ineligible abstract idea.” *W. Express Bancshares, LLC v. Green Dot Corp.*, 816 F. App’x 485, 487 (Fed. Cir. 2020).

Applying this fundamental economic practice to tangible components such as commercially available network equipment does not alter the fact that the claims are directed to “an art or principle in the abstract, and not for any particular method or machinery.” *ChargePoint, Inc. v. SemaConnect, Inc.*, 920 F.3d 759, 769-70 (Fed. Cir. 2019) (quoting *Wyeth v. Stone*, 30 F. Cas. 723 (C.C.D. Mass. 1840)) (“In short, the inventors here had the good idea to add networking capabilities to existing charging stations to facilitate various business interactions. But that is where they stopped, and that is all they patented. We therefore hold that claim 1 is ‘directed to’ an abstract idea”). The patentee’s implementation rests wholly on the existing knowledge of a person of skill in the art: “The Hotspot 10 comprises all the equipment installed at the rural location that is necessary to provide Internet access via satellite.” ’469 patent at 4:18-21 (emphasis added). “The Ethernet wireless access point 40 can employ a security protocol of a type that is known to persons skilled in the art, such as the 802.11b security protocol, or any subsequent or future versions of the 802.11 standard.” ’469 patent at 5:19-25 (emphasis added). The Federal Circuit has “repeatedly found the concept of controlling access to resources via software to be an abstract idea.” *Ericsson Inc. v. TCL Comm’n Tech. Holdings Ltd.*, 955 F.3d 1317, 1327 (Fed. Cir. 2020); see also *Smartflash LLC v. Apple Inc.*, 680 F. App’x 977 (Fed. Cir. 2017).

b) The Asserted Claims of the ’469 patent lack an inventive concept.

The specification provides no solution to overcome technical challenges associated with “remote”-ness, or a “high volume of transient traffic” or “subscriber access.” The claims of the Asserted Patent are results-driven and fail to overcome ineligibility. Each limitation recites

generic components which are “capable of” performing connectivity, routing, and authorization using any method whatsoever. As set forth above and in the attached claim charts, paid Internet hotspots were very well known and widely used at the time of the Asserted Patent. Placing a paid Internet hotspot in “a remote location” is insufficient to render the claims eligible where no particular improvement is recited to, for example, teach how a subscriber access unit is “capable of” authorizing payments in “a remote location.” *Universal Secure Registry LLC v. Apple Inc.*, 10 F.4th 1342, 1346 (Fed. Cir. 2021); *Elec. Power Grp., LLC v. Alstom S.A.*, 830 F.3d 1350, 1354 (Fed. Cir. 2016) (“limiting the claims to the particular technological environment of power-grid monitoring is, without more, insufficient to transform them into patent-eligible applications of the abstract idea at their core.”). The “essentially result-focused, functional character of claim language has been a frequent feature of claims held ineligible under § 101, especially in the area of using generic computer and network technology to carry out economic transactions.”

Each of the asserted dependent claims likewise recite the use of conventional components or processes to achieve results for which those components are designed. ’469 patent, claim 25 (“a wired data connection” “a wireless data connection”); claim 26 (“a plurality of users may access the Internet simultaneously,” “at the remote location,” “establishing data connections with the router via their web-ready devices”); claim 28 (“the data connections include wireless data connections”); claim 32 (“the wireless connection,” “802.11a wireless area network,” “802.11b wireless area network,” “802.11g wireless area network,” “802.11n wireless area network”). See also *CyberSource Corp. v. Retail Decisions, Inc.*, 654 F.3d 1366, 1370-73 (Fed. Cir. 2011) (a method of verifying the validity of credit card transactions over the Internet held patent-ineligible as directed to an abstract idea); *buySAFE, Inc. v. Google, Inc.*, 765 F.3d 1350, 1354-55 (Fed. Cir. 2014) (guaranteeing performance in an Internet transaction was patent-ineligible); *TLI*

Communications LLC Patent Litigation, 823 F.3d 607, 615 (Fed. Cir. 2016) (“vague, functional descriptions of server components are insufficient to transform the abstract idea into a patent-eligible invention.”). The claimed steps are described at a high level of generality, using generic terms, confirming the ’469 is ineligible under 35 U.S.C. §101.

G. Invalidity Based on Obviousness-Type Double Patenting

1. Obviousness-Type Double Patenting of the ’844 Patent in View of U.S. Patent No. 7,721,282

American provides further notice to Plaintiffs that the asserted claims of the ’844 patent are invalid under the doctrine of obviousness-type double patenting (“OTDP”) in view of claim 15 of U.S. Patent No. 7,721,282 (the “’282 patent”). Because the asserted claims of the ’844 patent are directed to the same invention recited by claims presented in the ’282 patent, and because any differences in the asserted claims of the ’844 patent are, at most, obvious modifications of claims in the ’282 patent, the asserted ’844 patent claims are invalid. *See AbbVie Inc. v. Mathilda & Terrence Kennedy Inst. of Rheumatology Tr.*, 764 F.3d 1366, 1373 (Fed. Cir. 2014).

The ’282 patent is available as an OTDP reference against the ’844 patent. The ’282 patent was filed on March 30, 2006, issued on May 18, 2010, and is set to expire on December 19, 2027. The ’844 patent was filed on February 21, 2007, issued on December 11, 2012, and is set to expire on April 6, 2028. Thus, the ’282 patent was filed earlier than the ’844 patent, issued earlier than the ’844 patent, and is set to expire earlier than the ’844 patent.

As is the case here, where “the applicant chooses to file separate applications for overlapping subject matter . . . the doctrine of obviousness-type double patenting ensures that a particular invention (and obvious variants thereof) does not receive an undue patent term

extension.” *AbbVie*, 764 F.3d at 1373. Therefore, the earlier-filed, earlier-issued, and earlier-expiring ’282 patent qualifies as a double patenting reference.

Given that the ’844 patent and the reference ’282 patent share at least one common inventor and common ownership, determining whether a patent claim is invalid for OTDP entails two steps. *See id.* at 1374. The first step determines the differences between the respective claims of the two patents. The second step determines whether those differences render the later-expiring set of claims sufficiently different to be patentably distinct. *Id.* “The second part of this analysis is analogous to the obviousness inquiry under 35 U.S.C. § 103 in the sense that if an earlier claim renders obvious or anticipates a later claim, the later claim is not patentably distinct and is thus invalid for obviousness-type double patenting.” *UCB, Inc. v. Accord Healthcare, Inc.*, 890 F.3d 1313, 1323 (Fed. Cir. 2018) (citation omitted). The disclosure of the reference patent may be used to determine whether the claims merely define an obvious variation of what is disclosed and claimed in the reference patent. *Abbvie*, 764 F.3d at 1381.

The attached claim chart in Exhibit 844-OTDP-1 demonstrates that, to the extent there are any differences between the asserted claims of the ’844 patent and claims of the ’282 patent, these differences are not patentably distinct. The charts provide reasons why a person of ordinary skill in the art would conclude that the alleged invention defined in the asserted claims of the ’844 patent would have been an obvious variation of the alleged invention defined in the corresponding claim or claims of the ’282 patent.

2. Obviousness-Type Double Patenting of the ’844 Patent in View of U.S. Patent No. 7,870,106

American provides further notice to Plaintiffs that the asserted claims of the ’844 patent are invalid under the doctrine of obviousness-type double patenting (“OTDP”) in view of claim 1 of U.S. Patent No. 7,870,106 (the “’106 patent”). Because the asserted claims of the ’844 patent

are directed to the same invention recited by claims presented in the '106 patent, and because any differences in the asserted claims of the '844 patent are, at most, obvious modifications of claims in the '106 patent, the asserted '844 patent claims are invalid. *See AbbVie Inc. v. Mathilda & Terrence Kennedy Inst. of Rheumatology Tr.*, 764 F.3d 1366, 1373 (Fed. Cir. 2014).

The '106 patent is available as an OTDP reference against the '844 patent. The '106 patent was filed on February 2, 2006, issued on January 11, 2011, and is set to expire on May 18, 2027. The '844 patent was filed on February 21, 2007, issued on December 11, 2012, and is set to expire on April 6, 2028. Thus, the '106 patent was filed earlier than the '844 patent, issued earlier than the '844 patent, and is set to expire earlier than the '844 patent.

As is the case here, where “the applicant chooses to file separate applications for overlapping subject matter . . . the doctrine of obviousness-type double patenting ensures that a particular invention (and obvious variants thereof) does not receive an undue patent term extension.” *AbbVie*, 764 F.3d at 1373. Therefore, the earlier-filed, earlier-issued, and earlier-expiring '106 patent qualifies as a double patenting reference.

Given that the '844 patent and the reference '106 patent share at least one common inventor and common ownership, determining whether a patent claim is invalid for OTDP entails two steps. *See id.* at 1374. The first step determines the differences between the respective claims of the two patents. The second step determines whether those differences render the later-expiring set of claims sufficiently different to be patentably distinct. *Id.* “The second part of this analysis is analogous to the obviousness inquiry under 35 U.S.C. § 103 in the sense that if an earlier claim renders obvious or anticipates a later claim, the later claim is not patentably distinct and is thus invalid for obviousness-type double patenting.” *UCB, Inc. v. Accord Healthcare, Inc.*, 890 F.3d 1313, 1323 (Fed. Cir. 2018) (citation omitted). The disclosure of the reference patent

may be used to determine whether the claims merely define an obvious variation of what is disclosed and claimed in the reference patent. *Abbvie*, 764 F.3d at 1381.

The attached claim chart in Exhibit 844-OTDP-2 demonstrates that, to the extent there are any differences between the asserted claims of the '844 patent and claims of the '106 patent, these differences are not patentably distinct. The charts provide reasons why a person of ordinary skill in the art would conclude that the alleged invention defined in the asserted claims of the '844 patent would have been an obvious variation of the alleged invention defined in the corresponding claim or claims of the '106 patent.

American reserves the right to assert any other double patenting challenges against any of the Asserted Patents.

III. ACCOMPANYING DOCUMENT PRODUCTION

Pursuant to P.R. 3-4(a), American is producing documentation in its possession that shows the operation of any aspects or elements of each Accused Instrumentality identified by IV in its P.R. 3-1(c) charts. *See, e.g.*, AA_IV_0002168 – AA_IV_0074538. However, American does not make any of the technologies accused of infringement. Moreover, for certain allegations, American has not identified any allegedly infringing use of the technologies based on IV's infringement allegations. American has collected certain source code in its possession relevant to the accused technologies and will make that code available pursuant to the source code provisions of the Protective Order, once entered by the Court. However, American does not represent that any of the code in its possession is relevant to IV's infringement allegations (to the extent that those allegations can be understood to identify an accused product).

Additionally, pursuant to P.R. 3-4(b), American is producing herewith a copy of each prior art reference identified above pursuant to P.R. 3-3(a) which does not appear in the file histories of the Asserted Patents, as well as additional prior art references. To the extent any such

prior art reference is not in English, American has produced to IV or will produce to IV an English translation of the portion(s) of the non-English prior art reference(s) relied upon by American.

Dated: June 20, 2025

/s/ John B. Campbell

John B. Campbell
Texas State Bar No. 24036314
jcampbell@McKoolSmith.com
Kenneth M. Scott
Texas State Bar No. 24137497
kscott@McKoolSmith.com
McKool Smith, P.C.
303 Colorado Street Suite 2100
Austin, TX 78701
Telephone: (512) 692-8700
Telecopier: (512) 692-8744

Emily Tannenbaum
New York State Bar No. 5928130
etannenbaum@mckoolsmith.com
McKool Smith, P.C.
1301 Avenue of the Americas, 32nd Floor
New York, NY 10019
Telephone: (212) 402-9400
Telecopier: (212) 402-9444

Casey L. Shomaker
Texas State Bar No. 24110359
cshomaker@mckoolsmith.com
McKool Smith, P.C.
300 Crescent Court, Suite 1500
Dallas, TX 75201
Telephone: (214) 978-4000
Facsimile: (214) 978-4044

Alan P. Block
California State Bar No. 143783
ablock@mckoolsmith.com
McKool Smith, P.C.
300 South Grand Avenue, Suite 2900
Los Angeles, CA 90071
Telephone: (213) 694-1054
Telecopier: (213) 694-1234

ATTORNEYS FOR DEFENDANT
AMERICAN AIRLINES, INC.

CERTIFICATE OF SERVICE

I hereby certify that a true and correct copy of the above and foregoing document has been served on all counsel of record via email on June 20, 2025.

/s/ John B. Campbell
John B. Campbell

Exhibit C-28

Invalidity of U.S. Patent No. 7,949,785 in View of Cisco VPN Routers and Software

Cisco VPN Routers and Software was known in this country, used in public, sold or offered for sale no later than 2002, as demonstrated by at least the materials cited herein or the testimony of knowledgeable witnesses and corroborating documents.¹ Specifically, based on information currently available, Cisco VPN Routers and Software was (1) known or used in this country before the alleged invention of the claimed subject matter of the asserted claims, (2) in public use or on sale in this country more than one year before the filing date of the patent, or (3) invented by another who did not abandon, suppress, or conceal, before the alleged invention of the claimed subject matter of the asserted claim. Thus, Cisco VPN Routers and Software anticipates the asserted claims under at least 35 U.S.C. §§ 102(a) or (g) as to claims 30, 35, and 37 (the “Asserted Claims”) of the ’785 Patent, as discussed in detail in the chart that follows.

At least the following documents describe the relevant functionality disclosed by Cisco VPN Routers and Software:

- Cisco 7100 Series VPN Routers;
- Cisco VPN 3000 Series Concentrator Data Sheet;
- CiscoWorks VPN/Security Management Solution 2.1;
- Cisco IPSec Encryption;
- Cisco 2600 and 3600 VPN Router Bundles;
- Cisco SEP Installation Guide;
- CiscoWorks VMS 2.2.

Based on information currently available, the persons or entities involved in the conception and diligent reduction to practice of the ideas in Cisco VPN Routers and Software include at least persons involved in the Cisco VPN Routers and Software and their testing and commercial use, including at least Cisco Systems. The conception of Cisco VPN Routers and Software occurred at least as early as 2002.

Defendant offers this invalidity chart based on its current understanding of Cisco VPN Routers and Software. Defendant’s investigation of this system is ongoing. Defendant may rely on materials produced by Plaintiffs or by third parties regarding Cisco

¹ Defendant’s invalidity charts, in some instances, rely at least in part on Plaintiff’s apparent positions regarding the scope of its claims for purposes of asserting infringement. Nothing in these claim charts should be understood as an admission relating to infringement, either literally or under the doctrine of equivalents, or as an admission relating to Defendant’s understanding of the proper interpretation or scope of the Asserted Claims. Defendant reserves the right to rely on additional citations or sources of evidence that may also be applicable, or that may become applicable in light of any Court Order on claim construction, changes in Plaintiff’s infringement contentions, or information obtained during discovery as the case proceeds.

Exhibit C-28

VPN Routers and Software and related software to demonstrate invalidity of the asserted claims. Defendant has not had an opportunity to review source code for Cisco VPN Routers and Software. Defendant may rely on source code, when available, implementing the functionality described herein, or other source code implementing substantially similar functionality, to demonstrate the invalidity of the asserted claims.

To the extent Plaintiff alleges Cisco VPN Routers and Software does not disclose any particular limitation of the Asserted Claims of the '785 Patent, either expressly or inherently, any purported differences are such that the claimed subject matter as a whole would have been obvious in view of the knowledge of one skilled in the art. It would have further been obvious to a person of ordinary skill in the art as of the priority date of the '785 Patent to modify Cisco VPN Routers and Software or combine the teachings of Cisco VPN Routers and Software with other prior art in a manner that would have rendered the Asserted Claims invalid as obvious, including but not limited to:

- U.S. Patent Application Publication No. 2003/0028671 (“Mehta”);
- Huitema, C., Network Working Group Request for Comment (RFC): 1383, entitled An Experiment in DNS Based IP Routing (“RFC 1383”);
- U.S. Patent No. 6,970,941 (“Caronni I”);
- U.S. Patent No. 7,814,228 (“Caronni II”); and/or
- U.S. Patent No. 6,766,371 (“Hipp”).

As the United States Supreme Court held in *KSR Int'l Co. v. Teleflex, Inc.*, “[t]he combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results.” 550 U.S. 398, 416 (2007). The Supreme Court further held: “[w]hen a work is available in one field of endeavor, design incentives and other market forces can prompt variations of it, either in the same field or a different one. If a person of ordinary skill can implement a predictable variation, § 103 likely bars its patentability. For the same reason, if a technique has been used to improve one device, and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond his or her skill.” *Id.* at 417. The Supreme Court further held: “[w]hen there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp. If this leads to the anticipated success, it is likely the product not of innovation but of ordinary skill and common sense.” *Id.* at 421.

Defendant reserves the right to amend or supplement this claim chart at a later date as more fully set forth in the Invalidity Contentions.

Exhibit C-28

Claims	Cisco VPN Routers and Software
<p>[30.pre] A virtual network manager, comprising:</p>	<p>To the extent the preamble is limiting, Cisco VPN Routers and Software discloses or renders obvious a virtual network manager. For example:</p> <p>“The Cisco 7100 Series VPN Router is a high-end, integrated VPN solution melding high-speed, industry-leading routing with a comprehensive suite of advanced site-to-site VPN services.” (Cisco 7100 Series VPN Routers).</p> <p>“The Cisco VPN 3000 Concentrator Series is a best-of-breed, remote-access VPN solution for enterprise-class deployment.” (Cisco VPN 3000 Series Concentrator Data Sheet).</p> <p>“CiscoWorks VMS provides Web-based tools for configuring, monitoring, and troubleshooting enterprise virtual private networks (VPNs), firewalls, and network host-based intrusion detection systems (IDSs).” (CiscoWorks VPN/Security Management Solution 2.1).</p> <p>To the extent this reference does not expressly or inherently disclose this limitation, it would have been obvious in view of the background knowledge of a person of ordinary skill in the art. This reference further renders the limitation obvious in light of other references, such as Mehta and/or Caronni I. Mehta and Caronni I concern similar subject matter as this reference, namely, virtual networks. As set forth in the accompanying contentions, use of virtual networks was well-known and common. Modifying this reference with Mehta’s and Caronni I’s teachings would have only required simple substitution, would not have required extensive experimentation, and would have yielded predictable results. A person of ordinary skill in the art would have further understood how to modify the disclosures of this reference with that of Mehta and Caronni I for example, to incorporate this limitation. <i>See, e.g.,</i></p> <p>“Methods and systems for providing two-way initiated, bi-directional communication with wireless devices using connection-based or connection-less protocols, such as, for example, TCP/IP and UDP/IP, are provided. Example</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>embodiments provide an Address Management Proxy System (“AMPS”), which enables devices and systems connected to a public internet, such as the Internet, to initiate communication with and to send data to wireless devices connected to a private wireless network, without exposing the non-routable private addresses of these wireless devices. The AMPS allocates a public (routable) network address for temporarily use by a requesting device on a public network to communicate with a wireless device on a wireless network. In one embodiment, a pool of public addresses, for example, public IP addresses, is maintained by the AMPS and allocated dynamically to wireless network devices as required. In one embodiment, the AMPS comprises one or more modified DNS/API servers, one or more Address Proxy/Routers, an Address Management Data Server, one or more data repositories, and optionally a load balancer. The AMPS DNS/API server receives a request from a device on a public network for a particular wireless device, and returns an appropriate temporary public address, which is internally mapped to the private address of the wireless device. The public address is then usable by the device on the public network to send data to the wireless device.” Mehta at Abstract.</p> <p>“The present invention relates to methods and systems for initiating communication with wireless devices and, in particular, to methods and systems for initiating communication with a device on a private network from a device on a public network to achieve virtual end-to-end connectivity.” Mehta at [0002].</p> <p>“Embodiments of the present invention provide computer- and network-based methods and systems for two-way initiated, bi-directional communication with wireless devices using connection-based or connection-less protocols, such as, for example, TCP/IP and UDP/IP. Example embodiments provide an Address Management Proxy System (“AMPS”), which enables devices and systems connected to a public internet network, such as the Internet, to initiate communication with and send data to wireless devices connected to a private wireless network, without exposing the non-routable private addresses of these wireless devices. The AMPS allocates a public (routable) network address for temporary use by a</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>requesting device on an external public network to communicate with a wireless device on a private wireless network. In one embodiment, a pool of public addresses, for example, public IP addresses, is maintained by the AMPS and allocated dynamically to wireless network devices as required. The mapping of a temporary public address to the private address of a wireless device is maintained and updated transparently by the AMPS using routing tables and other mapping data structures.” Mehta at [0011].</p> <p>“To insure a greater degree of security, according to one embodiment, the AMPS maintains a Time to Live (TTL) parameter with each address mapping. In this way, once the TTL value indicates that the mapping has expired, the AMPS can destroy the mapping, and also any connection. Further, in some embodiments, the AMPS also puts a timestamp in its own mapping tables. After some timeout period based upon the timestamp, the AMPS can destroy the mapping, thereby forcing a new mapping to be initiated on a periodic basis.” Mehta at [0015].</p> <p>“Embodiments of the present invention provide computer- and network-based methods and systems for two-way initiated, bi-directional communication with wireless devices using connection-based or connection-less protocols, such as, for example, TCP/IP and UDP/IP. Example embodiments provide an Address Management Proxy System (“AMPS”), which enables devices and systems connected to a public internet, such as the Internet, to initiate communication with and to send data to wireless devices connected to a private wireless network, without exposing the non-routable private addresses of these wireless devices.” Mehta at [0024].</p> <p>“The second approach used to implement the AMPS supports full bi-directional communication through point-to-point connections established, for example, using TCP/IP protocol. (Note that these same techniques also support connection-less UDP bi-directional communication). The second approach can be implemented by providing a modified implementation of a standard UDP or TCP/IP function, “GetHostByName.” The GetHostByName API allows a string designation to</p>

Exhibit C-28

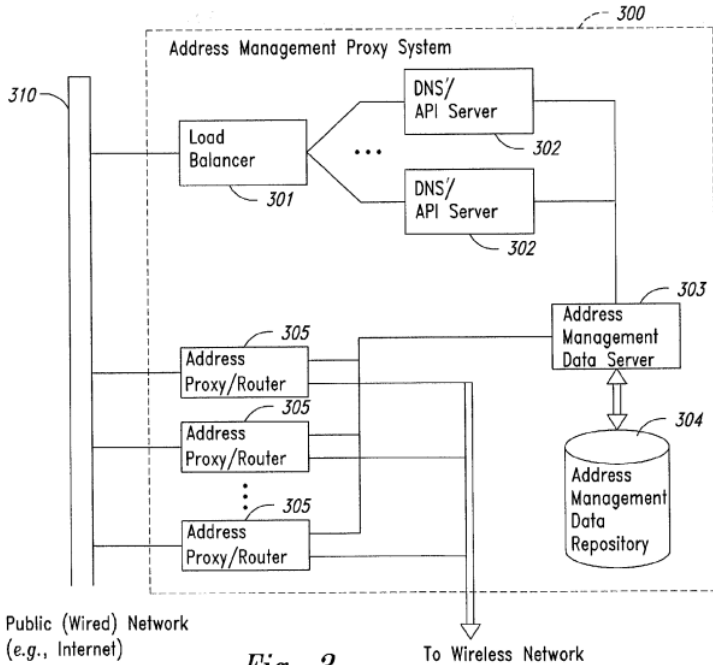
Claims	Cisco VPN Routers and Software
	<p>identify the designated device and returns an IP address data structure. Alternatively, to increase the level of security provided, the AMPS can implement a specialized API to return the dynamically allocated public address that (now) corresponds to the requested wireless device. A disadvantage of the specialized API approach is that the device on the public network (or other device that wishes to obtain a connection to the wireless device) needs to include specialized code in the application on the requesting device.” Mehta at [0041].</p>  <p style="text-align: center;"><i>Fig. 3</i></p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>Mehta at FIG. 3.</p> <p>“Methods and systems consistent with the present invention establish a virtual network on top of current IP network naming schemes. The virtual network uses a separate layer to create a modification to the IP packet format that is used to separate network behavior from addressing. As a result of the modification to the packet format, any type of delivery method may be assigned to any address or group of addresses. The virtual network also maintains secure communications between nodes, while providing the flexibility of assigning delivery methods independent of the delivery addresses.” Caronni I at Abstract.</p> <p>“The present invention relates generally to data processing systems and, more particularly, to a private network using a public-network infrastructure.” Caronni I at 1:57–59.</p> <p>“Methods and systems consistent with the present invention overcome the shortcomings of existing networks by establishing a “Supernet,” which is a private network that uses components from a public-network infrastructure. A Supernet allows an organization to utilize a public-network infrastructure for its enterprise network so that the organization no longer has to maintain a private network infrastructure; instead, the organization may have the infrastructure maintained for them by one or more service providers or other organizations that specialize in such connectivity matters. As such, the burden of maintaining an enterprise network is greatly reduced. Moreover, a Supernet is not geographically restrictive, so a user may plug their device into the Internet from virtually any portal in the world and still be able to use the resources of their private network in a secure and robust manner.” Caronni I at 4:36-52.</p> <p>“Supernets also provide heterogeneous addressing functionality. The Supernet uses a separate layer that isolates address names of nodes from addressing schemes and delivery schemes. The Supernet contains a modification to the IP packet format that can be used to separate network behavior from addressing. As a result of the</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>modification, any delivery scheme may be assigned to any address, or group of addresses.” Caronni I at 4:53-59.</p> <p><i>See also</i> Caronni I at Claim 1, FIGS. 3, 5.</p> <p>To the extent that Cisco VPN Routers and Software does not disclose any elements of this claim limitation, it would have been obvious to a person having ordinary skill in the art to combine, with the teachings of Cisco VPN Routers and Software, those elements, as known to them or as disclosed by other prior art identified in the cover pleading to Defendant’s Invalidation Contentions.</p>
<p>[30.1] a network interface configured for data communication via a virtual network that is defined by a domain name having an associated public network address;</p>	<p>Cisco VPN Routers and Software discloses or renders obvious a network interface configured for data communication via a virtual network that is defined by a domain name having an associated public network address. For example:</p> <p>"Dual autosensing 10/100BaseT Fast Ethernet ports." (Cisco 7100 Series VPN Routers)</p> <p>"Cisco VPN 3015-3080—Three auto-sensing, full-duplex 10/100BaseTX Fast Ethernet (public/untrusted, private/trusted and DMZ)." (Cisco VPN 3000 Series Concentrator Data Sheet)</p> <p>"Apply crypto map to interface. The commands below apply the crypto map to the interface." (Cisco IPSec Encryption)</p> <p>"Dual 10/100 Ethernet Router with 2 WIC Slots & 1 NM Slot." (Cisco 2600 and 3600 VPN Router Bundles)</p> <p>To the extent this reference does not expressly or inherently disclose this limitation, it would have been obvious in view of the background knowledge of a person of ordinary skill in the art. This reference further renders the limitation obvious in light of other references, such as Mehta and/or Caronni I. Mehta and Caronni I concern similar subject matter as this reference, namely, virtual networks. As set forth in the</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>accompanying contentions, use of virtual networks was well-known and common. Modifying this reference with Mehta’s and/or Caronni I’s teachings would have only required simple substitution, would not have required extensive experimentation, and would have yielded predictable results. A person of ordinary skill in the art would have further understood how to modify the disclosures of this reference with that of Mehta or Caronni I for example, to incorporate this limitation. <i>See, e.g.,</i></p> <p>“The present invention relates to methods and systems for initiating communication with wireless devices and, in particular, to methods and systems for initiating communication with a device on a private network from a device on a public network to achieve virtual end-to-end connectivity.” Mehta at [0002].</p> <p>“Embodiments of the present invention provide computer- and network-based methods and systems for two-way initiated, bi-directional communication with wireless devices using connection-based or connection-less protocols, such as, for example, TCP/IP and UDP/IP. Example embodiments provide an Address Management Proxy System (“AMPS”), which enables devices and systems connected to a public internet network, such as the Internet, to initiate communication with and send data to wireless devices connected to a private wireless network, without exposing the non-routable private addresses of these wireless devices. The AMPS allocates a public (routable) network address for temporary use by a requesting device on an external public network to communicate with a wireless device on a private wireless network. In one embodiment, a pool of public addresses, for example, public IP addresses, is maintained by the AMPS and allocated dynamically to wireless network devices as required. The mapping of a temporary public address to the private address of a wireless device is maintained and updated transparently by the AMPS using routing tables and other mapping data structures.” Mehta at [0011].</p> <p>“Although the techniques of the AMPS are generally applicable to any a wired device communicating with a wireless device, the phrase “public network” (or</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>“wired network”) is used generally to imply any type of internetworked environment including a public network or a backbone that is somewhere down the line connected to one or more private or public networks. In addition, although the examples described herein often refer to the Internet, one skilled in the art will recognize that the concepts and inventions described are applicable to other forms and embodiments of internetworking, including, for example ATM type networks. Thus, techniques of the present invention can also be used by one device on a first wireless network to communicate with another wireless device on a second network—each device ends up communicating with the Address Proxy/Router of the other network. This scenario is feasible because each wireless network (or its carrier infrastructure) is connected to a proxy/router that is also connected (via a public address) to a public network. In addition, although a public network is sometimes also referred to herein as a wired network, one skilled in the art will recognize that any network that exposes routable (public) addresses may be implied. Thus, a wireless network with unique public (and routable) address can also employ techniques of the present invention to perform bi-directional communication. Also, one skilled in the art will recognize that terms such as wireless device, phone, handheld, etc., are used interchangeably to indicate any type of wireless device that is capable of operating with the AMPS. In addition, terms may have alternate spellings which may or may not be explicitly mentioned, and one skilled in the art will recognize that all such variations of terms are intended to be included.” Mehta at [0033].</p> <p>“Public address to proxy/router machine table 620 comprises a public network address field 631 and an indication of a functioning proxy/router machine 621. By maintaining such a mapping, the AMPS is able to substitute proxy/router machines for other proxy/router machines to provide a higher degree of robustness. Each proxy/router machine has a preconfigured set of public network addresses, such as are typically configured by network cards inserted into the proxy/router machine. These address are allocated in a standard fashion through prior purchase or obtaining from an address authorizing authority, currently the Internet Corporation for Assigned Names and Numbers (ICANN). When a machine is inserted for use</p>

Exhibit C-28

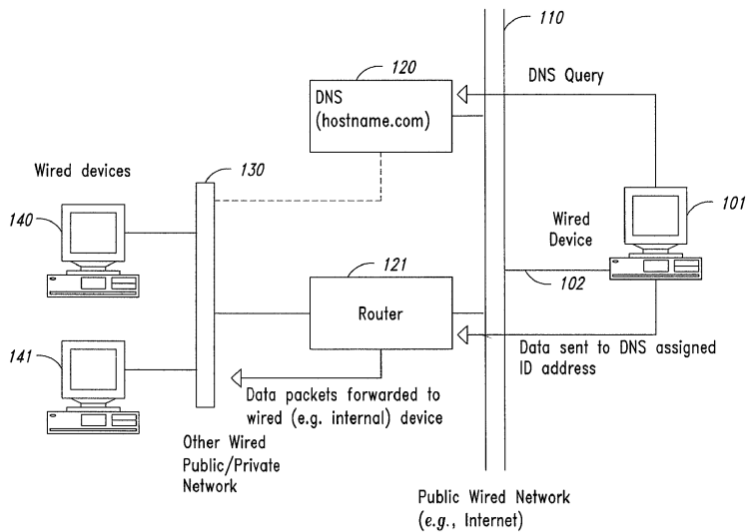
Claims	Cisco VPN Routers and Software
	<p>into the AMPS, indications of these addresses are stored in field 631.” Mehta at [0055].</p>  <p style="text-align: center;"><i>Fig. 1</i></p> <p>Mehta at FIG. 1.</p>

Exhibit C-28

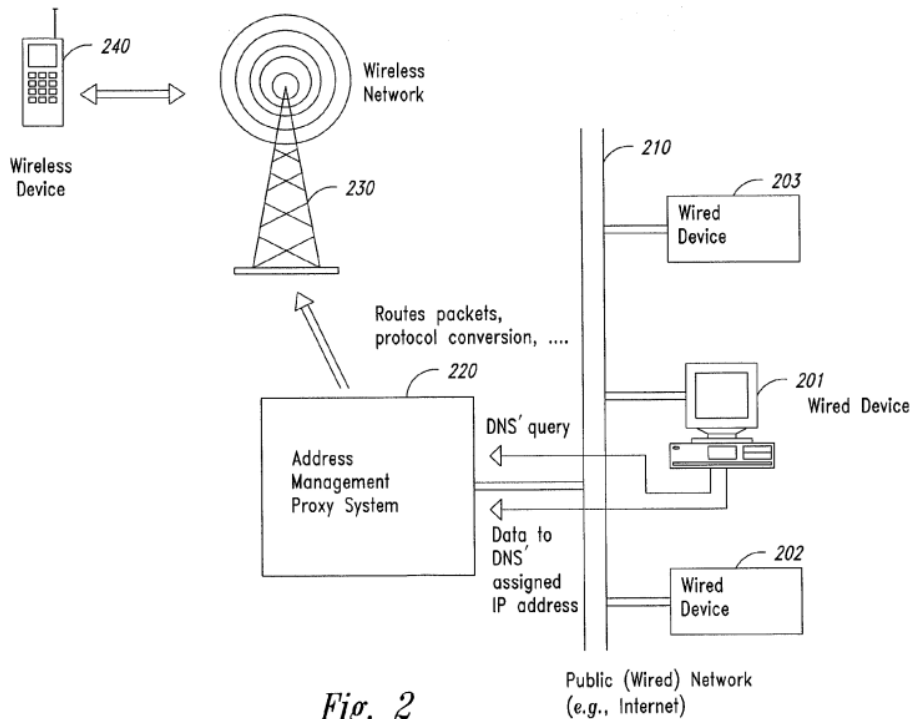
Claims	Cisco VPN Routers and Software
	 <p style="text-align: center;"><i>Fig. 2</i></p> <p>Mehta at FIG. 2.</p> <p>“Methods and systems consistent with the present invention establish a virtual network on top of current IP network naming schemes. The virtual network uses a separate layer to create a modification to the IP packet format that is used to separate network behavior from addressing. As a result of the modification to the packet format, any type of delivery method may be assigned to any address or group of addresses. The virtual network also maintains secure communications</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>between nodes, while providing the flexibility of assigning delivery methods independent of the delivery addresses.” Caronni I at Abstract.</p> <p>“The present invention relates generally to data processing systems and, more particularly, to a private network using a public-network infrastructure.” Caronni I at 1:57–59.</p> <p>“Methods and systems consistent with the present invention overcome the shortcomings of existing networks by establishing a “Supernet,” which is a private network that uses components from a public-network infrastructure. A Supernet allows an organization to utilize a public-network infrastructure for its enterprise network so that the organization no longer has to maintain a private network infrastructure; instead, the organization may have the infrastructure maintained for them by one or more service providers or other organizations that specialize in such connectivity matters. As such, the burden of maintaining an enterprise network is greatly reduced. Moreover, a Supernet is not geographically restrictive, so a user may plug their device into the Internet from virtually any portal in the world and still be able to use the resources of their private network in a secure and robust manner.” Caronni I at 4:36-52.</p> <p>“Supernets also provide heterogeneous addressing functionality. The Supernet uses a separate layer that isolates address names of nodes from addressing schemes and delivery schemes. The Supernet contains a modification to the IP packet format that can be used to separate network behavior from addressing. As a result of the modification, any delivery scheme may be assigned to any address, or group of addresses.” Caronni I at 4:53-59.</p> <p>“SNlogin 522 is a script used for logging into a Supernet. Successfully executing this script results in a Unix shell from which programs (e.g., node A 522) can be started to run within the Supernet context, such that address translation and security encapsulation is performed transparently for them and all they can typically access is other nodes on the Supernet. Alternatively, a parameter may be passed into</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>SNlogin 522 that indicates a particular process to be automatically run in a Supernet context. Once a program is running in a Supernet context, all programs spawned by that program also run in the Supernet context, unless explicitly stated otherwise. SNlogout 524 is a script used for logging out of a Supernet. Although both SNlogin 522 and SNlogout 524 are described as being scripts, one skilled in the art will appreciate that their processing may be performed by another form of software. VARPD 526 performs address translation between node IDs and real addresses. KMC 528 is the key management component for each node that receives updates whenever the key for a channel (“the channel key”) changes. There is one KMC per node per channel. KMD 530 receives requests from SNSL 542 of the TCP/IP protocol stack 534 when a packet is received and accesses the appropriate KMC for the destination node to retrieve the appropriate key to decrypt the packet. Node A 532 is a Supernet node running in a Supernet context.” Caronni I at 8:31-55.</p> <p>“TCP/IP protocol stack 534 contains a standard TCP/UDP layer 538, two standard IP layers (an inner IP layer 540 and an outer IP layer 544), and a Supernet security layer (SNSL) 542, acting as the conduit for all Supernet communications. To conserve memory, both inner IP layer 540 and outer IP layer 544 may share the same instance of the code of an IP layer. SNSL 542 performs security functionality as well as address translation. It also caches the most recently used channel keys for ten seconds. Thus, when a channel key is needed, SNSL 542 checks its cache first, and if it is not found, it requests KMD 530 to contact the appropriate KMC to retrieve the appropriate channel key.” Caronni I at 8:54-67.</p> <p>“FIG. 5 depicts administrative machine 306 and device 302 in greater detail, although the other devices 304 and 308–312 may contain similar components. Device 302 and administrative machine 306 communicate via Internet 314. Each device contains similar components, including a memory 502, 504; secondary storage 506, 508; a central processing unit (CPU) 510, 512; an input device 514, 516; and a video display 518, 520. One skilled in the art will appreciate that these devices may contain additional or different components. Memory 504 of</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>administrative machine 306 includes the SASD process 540, VARPD 548, and KMS 550 all running in user mode. That is, CPU 512 is capable of running in at least two modes: user mode and kernel mode. When CPU 512 executes programs running in user mode, it prevents them from directly manipulating the hardware components, such as video display 518. On the other hand, when CPU 512 executes programs running in kernel mode, it allows them to manipulate the hardware components. Memory 504 also contains a VARPDB 551 and a TCP/IP protocol stack 552 that are executed by CPU 512 running in kernel mode. TCP/IP protocol stack 552 contains a TCP/UDP layer 554 and an IP layer 556, both of which are standard layers well known to those of ordinary skill in the art. Secondary storage 508 contains a configuration file 558 that stores various configuration-related information (described below) for use by SASD 540.” Caronni I at 6:44-7:2.</p> <p><i>See also Caronni I at Claim 1, FIGS. 3, 5.</i></p> <p>To the extent that Cisco VPN Routers and Software does not disclose any elements of this claim limitation, it would have been obvious to a person having ordinary skill in the art to combine, with the teachings of Cisco VPN Routers and Software, those elements, as known to them or as disclosed by other prior art identified in the cover pleading to Defendant’s Invalidity Contentions.</p>
<p>[30.2] a memory and a processor to implement a register module configured to register devices in a virtual network,</p>	<p>Cisco VPN Routers and Software discloses or renders obvious a memory and a processor to implement a register module configured to register devices in a virtual network. For example:</p> <p>“Integrated Services Module (ISM) for hardware-based VPN services acceleration, such as high-speed IPsec or MPPE encryption.” (Cisco 7100 Series VPN Routers)</p> <p>“IPsec is the next-generation network layer crypto platform for Cisco's security platforms (Cisco IOS® Software, PIX, and so on).” (Cisco IPsec Encryption)</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>“Cisco SEP modules provide hardware-based encryption, ensuring consistent performance throughout the rated capacity.” (Cisco VPN 3000 Series Concentrator Data Sheet)</p> <p>“The VPN Concentrator automatically detects the new module during reboot.” (Cisco SEP Installation Guide)</p> <p>To the extent this reference does not expressly or inherently disclose this limitation, it would have been obvious in view of the background knowledge of a person of ordinary skill in the art. This reference further renders the limitation obvious in light of other references, such as Mehta, Caronni I, Caronni II, and/or Hipp. Mehta, Caronni I, Caronni II, and Hipp concern similar subject matter as this reference, namely, virtual networks. As set forth in the accompanying contentions, registration of devices in a virtual network was a well-known and commonly-used technique. Modifying this reference with Mehta’s, Caronni I’s, Caronni II’s, and/or Hipp’s teachings would have only required simple substitution, would not have required extensive experimentation, and would have yielded predictable results. A person of ordinary skill in the art would have further understood how to modify the disclosures of this reference with those of Mehta, Caronni I, Caronni II, or Hipp for example, to incorporate this limitation. <i>See, e.g.,</i></p> <p>“Methods and systems for providing two-way initiated, bi-directional communication with wireless devices using connection-based or connection-less protocols, such as, for example, TCP/IP and UDP/IP, are provided. Example embodiments provide an Address Management Proxy System (“AMPS”), which enables devices and systems connected to a public internet, such as the Internet, to initiate communication with and to send data to wireless devices connected to a private wireless network, without exposing the non-routable private addresses of these wireless devices. The AMPS allocates a public (routable) network address for temporarily use by a requesting device on a public network to communicate with a wireless device on a wireless network. In one embodiment, a pool of public</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>addresses, for example, public IP addresses, is maintained by the AMPS and allocated dynamically to wireless network devices as required. In one embodiment, the AMPS comprises one or more modified DNS/API servers, one or more Address Proxy/Routers, an Address Management Data Server, one or more data repositories, and optionally a load balancer. The AMPS DNS/API server receives a request from a device on a public network for a particular wireless device, and returns an appropriate temporary public address, which is internally mapped to the private address of the wireless device. The public address is then usable by the device on the public network to send data to the wireless device.” Mehta at Abstract.</p> <p>“To insure a greater degree of security, according to one embodiment, the AMPS maintains a Time to Live (TTL) parameter with each address mapping. In this way, once the TTL value indicates that the mapping has expired, the AMPS can destroy the mapping, and also any connection. Further, in some embodiments, the AMPS also puts a timestamp in its own mapping tables. After some timeout period based upon the timestamp, the AMPS can destroy the mapping, thereby forcing a new mapping to be initiated on a periodic basis.” Mehta at [0015].</p> <p>“In existing systems, data communication (communication on a data channel) between a wireless device that is connected to a private wireless network and a wired device connected to a public wired network (e.g., the Internet) can only be initiated by the wireless device. Some carriers have assigned fixed public IP addresses to wireless devices; however, the wireless devices need to then have client programs (e.g., a UDP stack) capability of receiving and handling the incoming communication packets. Moreover, these wireless devices are then part of a public wireless network and not a private wireless network. Since public IP addresses are becoming a scarcer commodity and currently expensive to service a network of multi-million devices, carriers cannot in a practical sense count on having a fixed public IP address for each device on its network. (Although movement to IPv6 will allow more addresses, the current IPv4 definitions are limited and the potential number of wireless users subscribing to larger carriers is</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>very high. In some regions of the world, the current public address scheme is even more limited.) Further, such addressing capability would expose each device to further security risks, because each such device is part of a publicly accessible network and it becomes more difficult to implement and enforce security measures. Thus, assigning private IP addresses to devices is preferred in existing wireless networks over assigning fixed public IP addresses to devices. When wireless networks are private, the locations (addresses) of the wireless devices are intentionally hidden from public view by a carrier system's (or, as referred to in some countries, operator's) infrastructure.” Mehta at [0025].</p> <p>“Example embodiments described herein provide applications, tools, data structures and other support to implement private to public address mappings over one or more wired and wireless networks to be used for bi-directional communication. One skilled in the art will recognize that other embodiments of the methods and systems of the present invention may be used for many other purposes, including to push information and/or data or code from a public network such as the Internet to a wireless device. In addition, although this description primarily refers to “data” as being sent via the networks, one skilled in the art will recognize that all types of data can be communicated using the techniques described herein including, but not limited to, text, graphics, audio, and video.” Mehta at [0034].</p> <p>“FIG. 4 is an example block diagram of a general purpose computer system for practicing embodiments of the Address Management Proxy System. The general purpose computer system 400 may comprise one or more server and/or client computing systems and may span distributed locations. In addition, each block may represent one or more such blocks as appropriate to a specific embodiment or may be combined with other blocks. The various blocks of the Address Management Proxy System 410 may physically reside on one or more machines, which use standard interprocess communication mechanisms to communicate with each other. In the embodiment shown, computer system 400 comprises a computer memory (“memory”) 01, a display 402, a Central Processing Unit (“CPU”) 403, and</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>Input/Output devices 404. The Address Management Proxy System 410 is shown residing in memory 401. The components of the Address Management Proxy System 410 preferably execute on CPU 403 and manage the address mapping of wireless devices on a wireless network, as described in previous figures, to allow other wired systems to communicate with the wireless devices. Other downloaded code 405 and potentially other data repositories also reside in the memory 410, and preferably execute on one or more CPU's 403. In a typical embodiment, the AMPS 410 includes one or more DNS/API servers 411, one or more Address Proxy/Routers 412, an Address Management Data Server 413, and Address Management Data Repositories 414. As described earlier, the AMPS may include other data repositories and components, such as a load balancer, depending upon the particular implementation.” Mehta at [0036].</p> <p>“In one embodiment, a timestamp of the mapping between a public network address and a private address is also noted in table 630. After a defined time out, based upon this timestamp, the Address Management Data Server sends a request to the Address Proxy/Router associated with the mapping to unmap the public-to-private address mapping and updates the mapping table 630, thereby returning the associated public address to the pool of unused public network addresses.” Mehta at [0056].</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<div data-bbox="919 289 1648 906" data-label="Diagram"> <p>The diagram, labeled Fig. 4, illustrates a Computer System (400). It features a central Memory block (401) which contains several sub-components: a DNS/API Server (411), an Address Proxy/Router (412), an Address Management Data Server (413), and an Address Management Data Repository (414). The repository is connected to a component labeled AMPS. To the right of the memory block is a block for Other Code (405). Below the memory block are three separate blocks: a Display (402), a CPU (403), and Input/Output Devices (404).</p> </div> <p data-bbox="1171 959 1283 1003"><i>Fig. 4</i></p> <p data-bbox="814 1062 1016 1089">Mehta at FIG. 4.</p> <p data-bbox="804 1159 1808 1326">“To configure a Supernet, a system administrator creates a configuration file 558 that is used by SASD 540 when starting or reconfiguring a Supernet. This file may specify: (1) the Supernet name, (2) all of the channels in the Supernet, (3) the nodes that communicate over each channel, (4) the address of the KMS for each channel, (5) the address of the VARPD that acts as the server for the Supernet, (6) the user</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>IDs of the users who are authorized to create Supernet nodes, (7) the authentication mechanism to use for each user of each channel, and (8) the encryption algorithm to use for each channel. Although the configuration information is described as being stored in a configuration file, one skilled in the art will appreciate that this information may be retrieved from other sources, such as databases or interactive configurations.” Caronni I at 8:1-15.</p> <p>“SASD 540 represents a Supernet: there is one instance of an SASD per Supernet, and it both authenticates nodes and authorizes nodes to join the Supernet. VARPD 548 has an associated component, VARPDB 551, into which it stores mappings of the internal Supernet addresses, known as a node IDs, to the network addresses recognized by the public-network infrastructure, known as the real addresses. The “node ID” may include the following: a Supernet ID (e.g., 0x123), reflecting a unique identifier of the Supernet, and a virtual address, comprising an IP address (e.g., 10.0.0.1). Although the virtual address is described in an IP address scheme, one skilled in the art will appreciate that the virtual address may be any other type addressing scheme, such as an e-mail address, IPX, or IPv6. Since the node ID includes a Supernet ID, a node will have more than one node ID when it communicates over more than one channel. The “real address” is an IP address (e.g., 10.0.0.2) that is globally unique and meaningful to the public-network infrastructure. In a Supernet, one VARPD runs on each machine, and it may play two roles. First, a VARPD may act as a server by storing all address mappings for a particular Supernet into its associated VARPDB. Second, regardless of its role as a server or not, each VARPD assists in address translation for the nodes on its machine. In this role, the VARPD stores into its associated VARPDB the address mappings for its nodes, and if it needs a mapping that it does not have, it will contact the VARPD that acts as the server for the given Supernet to obtain it. The VARPDB may also decide which virtual address to use in the translation. That is, the VARPDB may associate a virtual address with multiple real addresses or vice versa.” Caronni I at 7:3–33.</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>“FIGS. 7A and 7B depict a flow chart of the steps performed when a node joins a Supernet. The first step performed is that the user invokes the SNlogin script and enters the Supernet name, their user ID, their password, and a requested virtual address (step 702). Of course, this information depends on the particular authentication mechanism used. Upon receiving this information, the SNlogin script performs a handshaking with SASD to authenticate this information. In this step, the user may request a particular virtual address to be used, or alternatively, the SASD may select one for them. Next, if any of the information in step 702 is not validated by SASD (step 704), processing ends. Otherwise, upon successful authentication, SASD creates an address mapping between a node ID and the real address (step 706). In this step, SASD concatenates the Supernet ID with the virtual address to create the node ID, obtains the real address of the SNlogin script by querying network services in a well-known manner, and then registers this information with the VARPd that acts as the server for this Supernet. This VARPd is identified in the configuration file.” Caronni I at 9:67–10:18.</p> <p>“After configuring SNSL, SNlogin invokes an operating system call, SETVIN, to cause the SNlogin script to run in a Supernet context (step 720). In Unix, each process has a data structure known as the “proc structure” that contains the process ID as well as a pointer to a virtual memory description of this process. In accordance with methods and systems consistent with the present invention, the Supernet IDs indicating the channels over which the process communicates as well as its virtual address for this process are added to this structure. By associating this information with the process, the SNSL layer can enforce that this process runs in a Supernet context. Although methods and systems consistent with the present invention are described as operating in a Unix environment, one skilled in the art will appreciate that such methods and systems can operate in other environments. After the SNlogin script runs in the Supernet context, the SNlogin script spawns a Unix program, such as a Unix shell or a service daemon (step 722). In this step, the SNlogin script spawns a Unix shell from which programs can be run by the user. All of these</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>programs will thus run in the Supernet context until the user runs the SNlogout script.” Caronni I at 10:63–11:17.</p> <p>“The packet and Supernet ID are then transmitted to the SNSL layer using the modified socket structure (step 806). The SNSL layer then accesses the VARPDDB to obtain the address mapping between virtual source node address 642 and the source real address 614 as well as the virtual destination node address 644 and the destination real address 616 (step 808). If they are not contained in the VARPDDB because this is the first time a packet has been sent from this node or sent to this destination, the VARPDDB accesses the local VARPD to obtain the mapping. When contacted, the VARPD on the local machine contacts the VARPD that acts as the server for the Supernet to obtain the appropriate address mapping. Since the VARPDDB maintains all real IP addresses, a remote node may securely communicate with another remote node without reverfication.” Caronni I at 11:47–61.</p> <p><i>See also</i> Caronni I at FIGS. 7A and 7B.</p> <p>“In an embodiment of the present invention a virtual network is supported by a physical network and includes a virtual address resolution facility. The virtual address resolution facility is used to resolve a virtual IP address into a real IP address. A registration request is received at the virtual address resolution facility which includes a real IP address, a port address, a transport protocol designation and an Application layer protocol designation associated with a virtual address destination. The associations are stored using the virtual address resolution facility. A resolution request is received referencing a virtual address destination and the resolution request is resolved using the virtual address resolution facility and the stored associations.” Caronni II at 3:11–23.</p> <p>“FIG. 2 depicts a block diagram of the VARP lookup table 26 used by the virtual address resolution facility 24. The VARP lookup table 26 may be stored at any</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>location accessible over the network. The VARP lookup table 26 includes registered virtual destination addresses located in a column 78 and corresponding associations located in a column 79. Entries 80, 82, 84, 86 and 88 include the registered virtual destination address and information associated with the registered virtual destination address. Each entry in the VARP lookup table 26 will list the registering virtual IP address 90 and associated information including a real IP address 91, a transport protocol designation 92, a port number identification 93 and an Application layer protocol designation 94. The associations are provided at the time the virtual destination address is registered in the virtual network. For example, the entry 80 for the virtual IP address 10.0.0.12 (90) includes the association of a real IP address 152.70.0.1 (91), an associated Transport layer protocol designation, UDP 92, an associated port number, 6789 (93), and an Application layer protocol designation of ‘none’ 94.” Caronni II at 5:10–29.</p> <p>“FIG. 4 depicts the sequence of steps followed by the illustrative embodiment of the present invention to process the received encapsulated message. The sequence begins when a process associated with a virtual IP address registers with the virtual address resolution facility 24 (step 140). The registration is stored in a VARP lookup table accessible over the network. Subsequently, a message is received at the electronic device which bears a MAC/network interface address of the network interface of the electronic device with the designated real IP address listed in the virtual address registration (Step 142). The MAC address header 117 is stripped off at the Link layer and the message and appended headers are passed up to the Network layer 114(Step 144). The Network layer 114 strips off the IP header 115 and identifies the Transport Protocol header 113 underneath. The Network layer passes the message and remaining appended header to the Transport layer 112 (Step 146). The Transport layer 112 strips off the Transport header and then passes the message to the Application layer via the port address of the interface identified in the VARP registration. (Step 148). The receiving process determines whether it is the virtual destination address the message is addressed to (Step 149) and acts accordingly. If the receiving process that is examining the message is the destination</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>address, the process examines the data (Step 150). Alternatively, if the message is not intended for the process executing on the edge device, the process re-routes the message and sends it back down the protocol model stack, on to the local network operating behind the edge device, and on to its intended destination by performing traditional VARP and ARP lookups. Alternatively forwarding the message may be accomplished by reinserting the packet into the IP layer of the networking stack.” Caronni II at 7:33–67.</p> <p><i>See also</i> Caronni II at Claim 1, FIGS. 2-4.</p> <p>“FIG. 3 is a data flow diagram illustrating the registration of virtual network environment parameters. The VNE framework 200 is a software module that processes transactions between the applications and the operating system. The VNE parameters are registered with the VNE framework 200 at the time the application is started. The VNE parameters include the application IP address, the virtual network subnet, and the global virtual address subnet. At step 250, the registration harness 220 supplies the application IP address, virtual subnet, and the global virtual address subnet for a process_x to the VNE framework 200. The VNE framework 200 then records the IP address, virtual subnet, and the global virtual address subnet for the process_x at step 252. The process_x can then spawn additional processes (or create additional objects) at step 254. the new process_y inherits the IP address, virtual subnet, and global virtual address subnet from process_x. At step 256, the registration harness 220 launches the application related to process_y.” Hipp at 3:52-4:2.</p> <p>“Virtualization of network identity is achieved by assigning a unique virtual IP address and virtual hostname to a group of processes that make up the application instance which the instance keeps throughout its execution. This virtual network identity stays with the application instance regardless of which node the application is running on. The framework, in essence, provides a mechanism to create this virtual network identity (VNI) around the application using the virtual network</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>parameters assigned to it. In one embodiment, the virtual network parameters include an IP address and hostname. The framework 200 ensures that the application's instance uses the virtual network parameters, transparently, so that it can be moved across machines, without modifications to the application.” Hipp at 6:1-15.</p> <p><i>See also</i> Hipp at 6:16-60, FIG. 3.</p> <p>To the extent that Cisco VPN Routers and Software does not disclose any elements of this claim limitation, it would have been obvious to a person having ordinary skill in the art to combine, with the teachings of Cisco VPN Routers and Software, those elements, as known to them or as disclosed by other prior art identified in the cover pleading to Defendant’s Invalidation Contentions.</p>
<p>[30.3] the register module further configured to: receive a registration request from an agent associated with a device;</p>	<p>Cisco VPN Routers and Software discloses or renders obvious the register module further configured to: receive a registration request from an agent associated with a device. For example:</p> <p>“VPN access policies are created and stored centrally in the Cisco VPN 3000 Concentrator and pushed to the client when a connection is established.” (Cisco VPN 3000 Series Concentrator Data Sheet).</p> <p>“Centralized role-based Access Control (RBAC) enables organizations to scale access privileges.” (CiscoWorks VMS 2.2)</p> <p>To the extent this reference does not expressly or inherently disclose this limitation, it would have been obvious in view of the background knowledge of a person of ordinary skill in the art. This reference further renders the limitation obvious in light of other references, such as Mehta, Caronni I, Caronni II, and/or Hipp. Mehta, Caronni I, Caronni II, and Hipp concern similar subject matter as this reference, namely, virtual networks. As set forth in the accompanying contentions, registration of devices in a virtual network was a well-known and commonly-used technique. Modifying this</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>reference with Mehta’s, Caronni I’s, Caronni II’s, and/or Hipp’s teachings would have only required simple substitution, would not have required extensive experimentation, and would have yielded predictable results. A person of ordinary skill in the art would have further understood how to modify the disclosures of this reference with those of Mehta, Caronni I, Caronni II, or Hipp for example, to incorporate this limitation. <i>See, e.g.,</i></p> <p>“Methods and systems for providing two-way initiated, bi-directional communication with wireless devices using connection-based or connection-less protocols, such as, for example, TCP/IP and UDP/IP, are provided. Example embodiments provide an Address Management Proxy System (“AMPS”), which enables devices and systems connected to a public internet, such as the Internet, to initiate communication with and to send data to wireless devices connected to a private wireless network, without exposing the non-routable private addresses of these wireless devices. The AMPS allocates a public (routable) network address for temporarily use by a requesting device on a public network to communicate with a wireless device on a wireless network. In one embodiment, a pool of public addresses, for example, public IP addresses, is maintained by the AMPS and allocated dynamically to wireless network devices as required. In one embodiment, the AMPS comprises one or more modified DNS/API servers, one or more Address Proxy/Routers, an Address Management Data Server, one or more data repositories, and optionally a load balancer. The AMPS DNS/API server receives a request from a device on a public network for a particular wireless device, and returns an appropriate temporary public address, which is internally mapped to the private address of the wireless device. The public address is then usable by the device on the public network to send data to the wireless device.” Mehta at Abstract.</p> <p>“To insure a greater degree of security, according to one embodiment, the AMPS maintains a Time to Live (TTL) parameter with each address mapping. In this way, once the TTL value indicates that the mapping has expired, the AMPS can destroy</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>the mapping, and also any connection. Further, in some embodiments, the AMPS also puts a timestamp in its own mapping tables. After some timeout period based upon the timestamp, the AMPS can destroy the mapping, thereby forcing a new mapping to be initiated on a periodic basis.” Mehta at [0015].</p> <p>“In existing systems, data communication (communication on a data channel) between a wireless device that is connected to a private wireless network and a wired device connected to a public wired network (e.g., the Internet) can only be initiated by the wireless device. Some carriers have assigned fixed public IP addresses to wireless devices; however, the wireless devices need to then have client programs (e.g., a UDP stack) capability of receiving and handling the incoming communication packets. Moreover, these wireless devices are then part of a public wireless network and not a private wireless network. Since public IP addresses are becoming a scarcer commodity and currently expensive to service a network of multi-million devices, carriers cannot in a practical sense count on having a fixed public IP address for each device on its network. (Although movement to IPv6 will allow more addresses, the current IPv4 definitions are limited and the potential number of wireless users subscribing to larger carriers is very high. In some regions of the world, the current public address scheme is even more limited.) Further, such addressing capability would expose each device to further security risks, because each such device is part of a publicly accessible network and it becomes more difficult to implement and enforce security measures. Thus, assigning private IP addresses to devices is preferred in existing wireless networks over assigning fixed public IP addresses to devices. When wireless networks are private, the locations (addresses) of the wireless devices are intentionally hidden from public view by a carrier system's (or, as referred to in some countries, operator's) infrastructure.” Mehta at [0025].</p> <p>“Example embodiments described herein provide applications, tools, data structures and other support to implement private to public address mappings over one or more wired and wireless networks to be used for bi-directional communication. One skilled in the art will recognize that other embodiments of the</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>methods and systems of the present invention may be used for many other purposes, including to push information and/or data or code from a public network such as the Internet to a wireless device. In addition, although this description primarily refers to “data” as being sent via the networks, one skilled in the art will recognize that all types of data can be communicated using the techniques described herein including, but not limited to, text, graphics, audio, and video.” Mehta at [0034].</p> <p>“In one embodiment, a timestamp of the mapping between a public network address and a private address is also noted in table 630. After a defined time out, based upon this timestamp, the Address Management Data Server sends a request to the Address Proxy/Router associated with the mapping to unmap the public-to-private address mapping and updates the mapping table 630, thereby returning the associated public address to the pool of unused public network addresses.” Mehta at [0056].</p> <p>“FIG. 3 depicts a data processing system 300 suitable for use with methods and systems consistent with the present invention. Data processing system 300 comprises a number of devices, such as computers 302–312, connected to a public network, such as the Internet 314. A Supernet’s infrastructure uses components from the Internet because devices 302, 304, and 312 contain nodes that together form a Supernet and that communicate by using the infrastructure of the Internet. These nodes 316, 318, 320, and 322 are communicative entities (e.g., processes) running within a particular device and are able to communicate among themselves as well as access the resources of the Supernet in a secure manner. When communicating among themselves, the nodes 316, 318, 320, and 322 serve as end points for the communications, and no other processes or devices that are not part of the Supernet are able to communicate with the Supernet’s nodes or utilize the Supernet’s resources. The Supernet also includes an administrative node 306 to administer to the needs of the Supernet.” Caronni I at 4:66–5:13.</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>“FIG. 5 depicts administrative machine 306 and device 302 in greater detail, although the other devices 304 and 308–312 may contain similar components. Device 302 and administrative machine 306 communicate via Internet 314. Each device contains similar components, including a memory 502, 504; secondary storage 506, 508; a central processing unit (CPU) 510, 512; an input device 514, 516; and a video display 518, 520. One skilled in the art will appreciate that these devices may contain additional or different components. Memory 504 of administrative machine 306 includes the SASD process 540, VARPd 548, and KMS 550 all running in user mode. That is, CPU 512 is capable of running in at least two modes: user mode and kernel mode. When CPU 512 executes programs running in user mode, it prevents them from directly manipulating the hardware components, such as video display 518. On the other hand, when CPU 512 executes programs running in kernel mode, it allows them to manipulate the hardware components. Memory 504 also contains a VARPDB 551 and a TCP/IP protocol stack 552 that are executed by CPU 512 running in kernel mode. TCP/IP protocol stack 552 contains a TCP/UDP layer 554 and an IP layer 556, both of which are standard layers well known to those of ordinary skill in the art. Secondary storage 508 contains a configuration file 558 that stores various configuration-related information (described below) for use by SASD 540.” Caronni I at 6:43–7:2.</p> <p>“Memory 502 of device 302 contains SNlogin script 522, SNlogout script 524, VARPd 526, KMC 528, KMD 530, and node A 532, all running in user mode. Memory 502 also includes TCP/IP protocol stack 534 and VARPDB 536 running in kernel mode.” Caronni I at 8:26-30.</p> <p>“FIGS. 7A and 7B depict a flow chart of the steps performed when a node joins a Supernet. The first step performed is that the user invokes the SNlogin script and enters the Supernet name, their user ID, their password, and a requested virtual address (step 702). Of course, this information depends on the particular authentication mechanism used. Upon receiving this information, the SNlogin script performs a handshaking with SASD to authenticate this information. In this</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>step, the user may request a particular virtual address to be used, or alternatively, the SASD may select one for them. Next, if any of the information in step 702 is not validated by SASD (step 704), processing ends. Otherwise, upon successful authentication, SASD creates an address mapping between a node ID and the real address (step 706). In this step, SASD concatenates the Supernet ID with the virtual address to create the node ID, obtains the real address of the SNlogin script by querying network services in a well-known manner, and then registers this information with the VARP that acts as the server for this Supernet. This VARP is identified in the configuration file.” Caronni I at 9:66-10:18.</p> <p><i>See also</i> Caronni I at FIGS. 3, 5, 7A, 7B.</p> <p>“In an embodiment of the present invention a virtual network is supported by a physical network and includes a virtual address resolution facility. The virtual address resolution facility is used to resolve a virtual IP address into a real IP address. A registration request is received at the virtual address resolution facility which includes a real IP address, a port address, a transport protocol designation and an Application layer protocol designation associated with a virtual address destination. The associations are stored using the virtual address resolution facility. A resolution request is received referencing a virtual address destination and the resolution request is resolved using the virtual address resolution facility and the stored associations.” Caronni II at 3:11–23.</p> <p>“FIG. 2 depicts a block diagram of the VARP lookup table 26 used by the virtual address resolution facility 24. The VARP lookup table 26 may be stored at any location accessible over the network. The VARP lookup table 26 includes registered virtual destination addresses located in a column 78 and corresponding associations located in a column 79. Entries 80, 82, 84, 86 and 88 include the registered virtual destination address and information associated with the registered virtual destination address. Each entry in the VARP lookup table 26 will list the registering virtual IP address 90 and associated information including a real IP address 91, a transport protocol designation 92, a port number identification 93 and</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>an Application layer protocol designation 94. The associations are provided at the time the virtual destination address is registered in the virtual network. For example, the entry 80 for the virtual IP address 10.0.0.12 (90) includes the association of a real IP address 152.70.0.1 (91), an associated Transport layer protocol designation, UDP 92, an associated port number, 6789 (93), and an Application layer protocol designation of ‘none’ 94.” Caronni II at 5:10–29.</p> <p>“FIG. 4 depicts the sequence of steps followed by the illustrative embodiment of the present invention to process the received encapsulated message. The sequence begins when a process associated with a virtual IP address registers with the virtual address resolution facility 24 (step 140). The registration is stored in a VARP lookup table accessible over the network. Subsequently, a message is received at the electronic device which bears a MAC/network interface address of the network interface of the electronic device with the designated real IP address listed in the virtual address registration (Step 142). The MAC address header 117 is stripped off at the Link layer and the message and appended headers are passed up to the Network layer 114(Step 144). The Network layer 114 strips off the IP header 115 and identifies the Transport Protocol header 113 underneath. The Network layer passes the message and remaining appended header to the Transport layer 112 (Step 146). The Transport layer 112 strips off the Transport header and then passes the message to the Application layer via the port address of the interface identified in the VARP registration. (Step 148). The receiving process determines whether it is the virtual destination address the message is addressed to (Step 149) and acts accordingly. If the receiving process that is examining the message is the destination address, the process examines the data (Step 150). Alternatively, if the message is not intended for the process executing on the edge device, the process re-routes the message and sends it back down the protocol model stack, on to the local network operating behind the edge device, and on to its intended destination by performing traditional VARP and ARP lookups. Alternatively forwarding the message may be accomplished by reinserting the packet into the IP layer of the networking stack.” Caronni II at 7:33–67.</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p><i>See also</i> Caronni II at Claim 1, FIGS. 2-4.</p> <p>“FIG. 3 is a data flow diagram illustrating the registration of virtual network environment parameters. The VNE framework 200 is a software module that processes transactions between the applications and the operating system. The VNE parameters are registered with the VNE framework 200 at the time the application is started. The VNE parameters include the application IP address, the virtual network subnet, and the global virtual address subnet. At step 250, the registration harness 220 supplies the application IP address, virtual subnet, and the global virtual address subnet for a processx to the VNE framework 200. The VNE framework 200 then records the IP address, virtual subnet, and the global virtual address subnet for the processx at step 252. The processx can then spawn additional processes (or create additional objects) at step 254. the new processy inherits the IP address, virtual subnet, and global virtual address subnet from processx. At step 256, the registration harness 220 launches the application related to processy.” Hipp at 3:52-4:2.</p> <p>“Virtualization of network identity is achieved by assigning a unique virtual IP address and virtual hostname to a group of processes that make up the application instance which the instance keeps throughout its execution. This virtual network identity stays with the application instance regardless of which node the application is running on. The framework, in essence, provides a mechanism to create this virtual network identity (VNI) around the application using the virtual network parameters assigned to it. In one embodiment, the virtual network parameters include an IP address and hostname. The framework 200 ensures that the application's instance uses the virtual network parameters, transparently, so that it can be moved across machines, without modifications to the application.” Hipp at 6:1-15.</p> <p><i>See also</i> Hipp at 6:16-60, FIG. 3.</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>To the extent that Cisco VPN Routers and Software does not disclose any elements of this claim limitation, it would have been obvious to a person having ordinary skill in the art to combine, with the teachings of Cisco VPN Routers and Software, those elements, as known to them or as disclosed by other prior art identified in the cover pleading to Defendant's Invalidation Contentions.</p>
<p>[30.4] distribute a virtual network address to the device when the device is registered in the virtual network, the device being identified to other devices in the virtual network by the virtual network address; and</p>	<p>Cisco VPN Routers and Software discloses or renders obvious the register module further configured to: distribute a virtual network address to the device when the device is registered in the virtual network, the device being identified to other devices in the virtual network by the virtual network address. For example:</p> <p>“Utilizing ISM and ISA hardware encryption acceleration, the Cisco 7100 Series VPN Router can support up to 3000 simultaneous IPsec tunneling sessions with 3DES IPsec encryption performance up to 140 Mbps.” (Cisco 7100 Series VPN Routers).</p> <p>“The Cisco VPN 3080 Concentrator is optimized to support large enterprise organizations that demand the highest level of performance combined with support for up to 10,000 simultaneous remote access sessions.” (Cisco VPN 3000 Series Concentrator Data Sheet).</p> <p>To the extent this reference does not expressly or inherently disclose this limitation, it would have been obvious in view of the background knowledge of a person of ordinary skill in the art. This reference further renders the limitation obvious in light of other references, such as Mehta, Caronni I, Caronni II and/or Hipp. Mehta, Caronni I, Caronni II and Hipp concern similar subject matter as this reference, namely, virtual networks. As set forth in the accompanying contentions, distributing virtual network addresses was a well-known and commonly-used technique. Modifying this reference with Mehta's, Caronni I's, Caronni II's, and/or Hipp's teachings would have only required simple substitution, would not have required extensive experimentation, and would have yielded predictable results. A person of ordinary skill in the art would have further</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>understood how to modify the disclosures of this reference with those of Mehta, Caronni I, Caronni II, or Hipp for example, to incorporate this limitation. <i>See, e.g.,</i></p> <p>“Methods and systems for providing two-way initiated, bi-directional communication with wireless devices using connection-based or connection-less protocols, such as, for example, TCP/IP and UDP/IP, are provided. Example embodiments provide an Address Management Proxy System (“AMPS”), which enables devices and systems connected to a public internet, such as the Internet, to initiate communication with and to send data to wireless devices connected to a private wireless network, without exposing the non-routable private addresses of these wireless devices. The AMPS allocates a public (routable) network address for temporarily use by a requesting device on a public network to communicate with a wireless device on a wireless network. In one embodiment, a pool of public addresses, for example, public IP addresses, is maintained by the AMPS and allocated dynamically to wireless network devices as required. In one embodiment, the AMPS comprises one or more modified DNS/API servers, one or more Address Proxy/Routers, an Address Management Data Server, one or more data repositories, and optionally a load balancer. The AMPS DNS/API server receives a request from a device on a public network for a particular wireless device, and returns an appropriate temporary public address, which is internally mapped to the private address of the wireless device. The public address is then usable by the device on the public network to send data to the wireless device.” Mehta at Abstract.</p> <p>“To insure a greater degree of security, according to one embodiment, the AMPS maintains a Time to Live (TTL) parameter with each address mapping. In this way, once the TTL value indicates that the mapping has expired, the AMPS can destroy the mapping, and also any connection. Further, in some embodiments, the AMPS also puts a timestamp in its own mapping tables. After some timeout period based</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>upon the timestamp, the AMPS can destroy the mapping, thereby forcing a new mapping to be initiated on a periodic basis.” Mehta at [0015].</p> <p>“In existing systems, data communication (communication on a data channel) between a wireless device that is connected to a private wireless network and a wired device connected to a public wired network (e.g., the Internet) can only be initiated by the wireless device. Some carriers have assigned fixed public IP addresses to wireless devices; however, the wireless devices need to then have client programs (e.g., a UDP stack) capability of receiving and handling the incoming communication packets. Moreover, these wireless devices are then part of a public wireless network and not a private wireless network. Since public IP addresses are becoming a scarcer commodity and currently expensive to service a network of multi-million devices, carriers cannot in a practical sense count on having a fixed public IP address for each device on its network. (Although movement to IPv6 will allow more addresses, the current IPv4 definitions are limited and the potential number of wireless users subscribing to larger carriers is very high. In some regions of the world, the current public address scheme is even more limited.) Further, such addressing capability would expose each device to further security risks, because each such device is part of a publicly accessible network and it becomes more difficult to implement and enforce security measures. Thus, assigning private IP addresses to devices is preferred in existing wireless networks over assigning fixed public IP addresses to devices. When wireless networks are private, the locations (addresses) of the wireless devices are intentionally hidden from public view by a carrier system's (or, as referred to in some countries, operator's) infrastructure.” Mehta at [0025].</p> <p>“FIG. 3 is an example block diagram of components of an example Address Management Proxy System. In one embodiment, the Address Management Proxy System (AMPS) comprises one or more modified DNS'/ API servers 302, one or more Address Proxy/Routers 305, an Address Management Data Server 303 which manages a database or other repositories such as Address Management Data Repository 304, and optionally a load balancer 301. The DNS'/API servers 302 are</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>either individually connected to a public network 310 or are connected to the load balancer 301 which is in turn connected to public network 310. Similarly, each Address Proxy/Router 305 is also connected to the public network 310, via the routable (public) address to which data from the external network 310 is sent that is destined for the wireless devices. The DNS'/API servers 302 are either modified implementations of a DNS server to add functionality necessary to communicate with wireless devices, or are servers that implement one or more specialized APIs, as will be described further below. The DNS'/API servers 302 use the Address Management Data Server 303 to assist in mapping a unique identifier (e.g., string name) for a wireless device to a public address on public network 310. The pool of public addresses is also maintained by the Address Management Data Serve 303 and Data Repository 304. The Address Management Data server 303 and Address Management Data Repository 304 may be implemented using existing database technology, for example, ODBC technology or, may be implemented as a structure such as a simple text file. One skilled in the art will recognize that any embodiment for storing a set of tables, data, lists, or mappings can be used. Each Address Proxy/Router 305 also uses the Address Management Data Server 303 or equivalent to create and update a series of routing tables that are used to assign public addresses to wireless devices as needed and to update the various mappings between public addresses and the non-routable (private) addresses of the wireless devices. The tables and mappings that are maintained on behalf of the DNS'/API servers 302 and the Address Proxy/Routers 305 by the Address Management Data Server 303 are described below with reference to FIG. 6.” Mehta at [0032].</p> <p>“Although the techniques of the AMPS are generally applicable to any a wired device communicating with a wireless device, the phrase “public network” (or “wired network”) is used generally to imply any type of internetworked environment including a public network or a backbone that is somewhere down the line connected to one or more private or public networks. In addition, although the examples described herein often refer to the Internet, one skilled in the art will recognize that the concepts and inventions described are applicable to other forms and embodiments of internetworking, including, for example ATM type networks.</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>Thus, techniques of the present invention can also be used by one device on a first wireless network to communicate with another wireless device on a second network—each device ends up communicating with the Address Proxy/Router of the other network. This scenario is feasible because each wireless network (or its carrier infrastructure) is connected to a proxy/router that is also connected (via a public address) to a public network. In addition, although a public network is sometimes also referred to herein as a wired network, one skilled in the art will recognize that any network that exposes routable (public) addresses may be implied. Thus, a wireless network with unique public (and routable) address can also employ techniques of the present invention to perform bi-directional communication. Also, one skilled in the art will recognize that terms such as wireless device, phone, handheld, etc., are used interchangeably to indicate any type of wireless device that is capable of operating with the AMPS. In addition, terms may have alternate spellings which may or may not be explicitly mentioned, and one skilled in the art will recognize that all such variations of terms are intended to be included.” Mehta at [0033].</p> <p>“Example embodiments described herein provide applications, tools, data structures and other support to implement private to public address mappings over one or more wired and wireless networks to be used for bi-directional communication. One skilled in the art will recognize that other embodiments of the methods and systems of the present invention may be used for many other purposes, including to push information and/or data or code from a public network such as the Internet to a wireless device. In addition, although this description primarily refers to “data” as being sent via the networks, one skilled in the art will recognize that all types of data can be communicated using the techniques described herein including, but not limited to, text, graphics, audio, and video.” Mehta at [0034].</p> <p>“There are several implementation approaches to the components of the Address Management Proxy System, three of which are described herein. One skilled in the art will recognize that various other approaches and combinations are possible. All</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>three approaches allocate a public (routable) network address for temporary use by a wired device to communicate with a wireless device. In one embodiment, a pool of public addresses, for example, public IP addresses, is maintained and allocated dynamically to wireless network devices as required. For example, a typical Class B Internet network address block allows for approximately 65,000 simultaneous connections to wireless devices. Although this number may seem large at first blush, when one considers the number of cell phones and handsets, for example, connected to a carrier, this number can be quite limiting.” Mehta at [0039].</p> <p>“The unique ID table 610 maps unique string names of wireless devices to the private wireless network addresses that have been assigned typically by a carrier's infrastructure. In some embodiments, the carrier infrastructure dynamically allocates, using methods similar to a DHCP protocol, a private wireless network address when the wireless device registers itself with the carrier infrastructure upon being powered on. Thus, the unique ID table 610 may be sparsely filled or entries created dynamically and then deleted dynamically as devices register and unregister with the carrier infrastructure system.” Mehta at [0053].</p> <p>“In one embodiment, a timestamp of the mapping between a public network address and a private address is also noted in table 630. After a defined time out, based upon this timestamp, the Address Management Data Server sends a request to the Address Proxy/Router associated with the mapping to unmap the public-to-private address mapping and updates the mapping table 630, thereby returning the associated public address to the pool of unused public network addresses.” Mehta at [0056].</p> <p>“FIG. 7 is an example flow diagram of an example routine provided by a DNS/API server of the Address Management Proxy System to return a public address that corresponds to a designated unique identifier. In essence, this routine implements a DNS query or DNS-like query capability for the AMPS using a modified GetHostByName interface or a specialized API, for example GetProxyIP, as described with reference to FIG. 5. In summary, the routine dynamically allocates</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>an appropriate Address Proxy/Router machine to associate with the wireless device and returns the public address of that machine (along with a TTL parameter and potentially other parameters). Specifically, in step 701, the routine determines the private (non-routable) address of the wireless device designated by a string parameter passed as input to the routine. For example, the string parameter may use fields such as “uniqueID.hostname.domain.tld,” which specifies a typical hierarchy of person/service on a machine named “hostname” on a domain such as a company’s network on a top level domain such as “org.” “com,” “edu,” etc. One skilled in the art will recognize that many other string parameter designations could be used. One mechanism for implementing this routine is to request information from the Address Management Data Repository. In one embodiment, the data repository stores a table that maps unique IDs to private network addresses (see Table 610 in FIG. 6). Preferably, any mechanism that is used by the AMPS stores this data in a secure manner in order to keep the wireless network addresses private. In step 702, the routine retrieves the public network address that corresponds to the private wireless network address of the designated device if one has already been assigned by the AMPS to that device and is still valid. In one embodiment, the data repository stores this mapping information between private wireless network address and public network address (see for example table 630 in FIG. 6). If a public network address has not already been assigned or is not valid, then the routine causes a new public network address to be allocated and that new public address is associated with the private wireless network address. Appropriate tables in the data repository are then updated. In step 703, the routine determines the Address Proxy/Router machine that is associated with the assigned public network address (for example using table 620 in FIG. 6). In step 704, the routine sends a request to the determined proxy/router machine to update its routing tables to map the determined public network address to the private wireless network address. In step 705, the routine updates information in the data repository to include any other connection related information (e.g., field 634 in Table 630 in FIG. 6) and indicates a Time to Live (TTL) parameter for the public-private address association (e.g., field 633 in Table 630 in FIG. 6). Once all of the tables have been updated in both</p>

Exhibit C-28

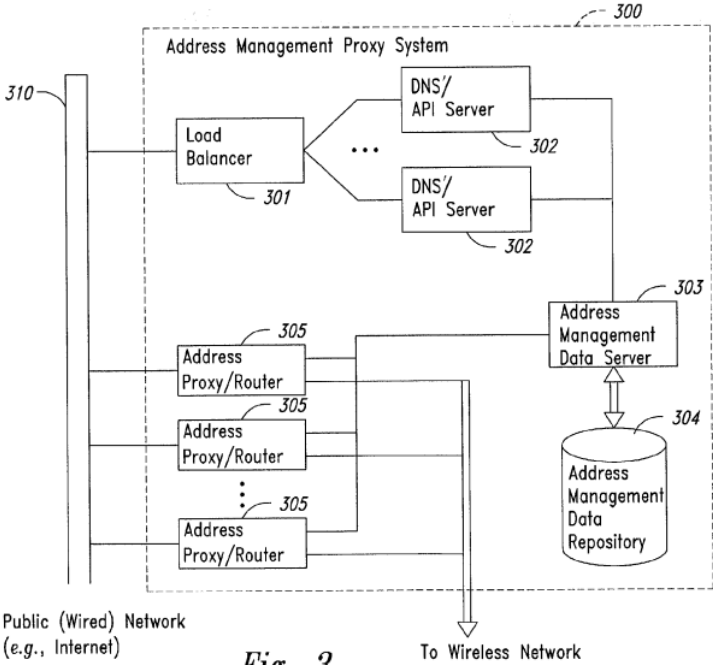
Claims	Cisco VPN Routers and Software
	<p>the proxy/router and in the data repository, the DNS'/API server returns the determined public network address of the associated Address Proxy/Router machine. As described earlier, the public address may be a (hostname, port) pair when a port-based implementation is used.” Mehta at [0057].</p>  <p style="text-align: center;"><i>Fig. 3</i></p> <p>Mehta at FIG. 3.</p> <p>“Fourth, the system provides address translation in a transparent manner. Since the Supernet is a private network constructed from the infrastructure of another</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>network, the Supernet has its own internal addressing scheme, separate from the addressing scheme of the underlying public network. Thus, when a packet from a Supernet node is sent to another Supernet node, it travels through the public network. To do so, the Supernet performs address translation from the internal addressing scheme to the public addressing scheme and vice versa. By separating the addressing schemes, the Supernet creates a flexible delivery scheme that is easily changeable by network software or a system administrator. To reduce the complexity of Supernet nodes, system-level components of the Supernet perform this translation on behalf of the individual nodes so that it is transparent to the nodes. Another benefit of the Supernet's addressing is that it uses an IP-based internal addressing scheme so that preexisting programs require little modification to run within a Supernet.” Caronni I at 6:7–25.</p> <p>“SASD 540 represents a Supernet: there is one instance of an SASD per Supernet, and it both authenticates nodes and authorizes nodes to join the Supernet. VARPD 548 has an associated component, VARPDB 551, into which it stores mappings of the internal Supernet addresses, known as a node IDs, to the network addresses recognized by the public-network infrastructure, known as the real addresses. The “node ID” may include the following: a Supernet ID (e.g., 0x123), reflecting a unique identifier of the Supernet, and a virtual address, comprising an IP address (e.g., 10.0.0.1). Although the virtual address is described in an IP address scheme, one skilled in the art will appreciate that the virtual address may be any other type addressing scheme, such as an e-mail address, IPX, or IPv6. Since the node ID includes a Supernet ID, a node will have more than one node ID when it communicates over more than one channel. The “real address” is an IP address (e.g., 10.0.0.2) that is globally unique and meaningful to the public-network infrastructure. In a Supernet, one VARPD runs on each machine, and it may play two roles. First, a VARPD may act as a server by storing all address mappings for a particular Supernet into its associated VARPDB. Second, regardless of its role as a server or not, each VARPD assists in address translation for the nodes on its machine. In this role, the VARPD stores into its associated VARPDB the address mappings for its nodes, and if it needs a mapping that it does not have, it will</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>contact the VARPDB that acts as the server for the given Supernet to obtain it. The VARPDB may also decide which virtual address to use in the translation. That is, the VARPDB may associate a virtual address with multiple real addresses or vice versa.” Caronni I at 7:3–33.</p> <p>“FIGS. 7A and 7B depict a flow chart of the steps performed when a node joins a Supernet. The first step performed is that the user invokes the SNlogin script and enters the Supernet name, their user ID, their password, and a requested virtual address (step 702). Of course, this information depends on the particular authentication mechanism used. Upon receiving this information, the SNlogin script performs a handshaking with SASD to authenticate this information. In this step, the user may request a particular virtual address to be used, or alternatively, the SASD may select one for them. Next, if any of the information in step 702 is not validated by SASD (step 704), processing ends. Otherwise, upon successful authentication, SASD creates an address mapping between a node ID and the real address (step 706). In this step, SASD concatenates the Supernet ID with the virtual address to create the node ID, obtains the real address of the SNlogin script by querying network services in a well-known manner, and then registers this information with the VARPDB that acts as the server for this Supernet. This VARPDB is identified in the configuration file.” Caronni I at 9:66-10:18.</p> <p>“Once inner IP layer 542 receives the packet, a Supernet ID is appended to a socket structure (step 804). The socket structure is modified so as to contain an extra data field for Supernet ID 626 and virtual source address 642. The addition of Supernet ID 626 and virtual address 642 in the socket structure enables the Supernet to communicate with nodes regardless of the delivery scheme used. When the process on node A opens a socket to transmit the packet to inner IP layer 540, the corresponding Supernet ID 626 and virtual source address 642 for that process is included in the socket request.</p> <p>The packet and Supernet ID are then transmitted to the SNSL layer using the modified socket structure (step 806). The SNSL layer then accesses the VARPDB</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>to obtain the address mapping between virtual source node address 642 and the source real address 614 as well as the virtual destination node address 644 and the destination real address 616 (step 808). If they are not contained in the VARPDB because this is the first time a packet has been sent from this node or sent to this destination, the VARPDB accesses the local VARPDB to obtain the mapping. When contacted, the VARPDB on the local machine contacts the VARPDB that acts as the server for the Supernet to obtain the appropriate address mapping. Since the VARPDB maintains all real IP addresses, a remote node may securely communicate with another remote node without reverfication.” Caronni I at 11:37–61.</p> <p><i>See also</i> Caronni I at FIGS. 7A and 7B.</p> <p>“Unfortunately, when the virtual destination address is located on a physical device on the interior of a network which is running a proxy server, firewall, or other packet filtering mechanism, messages that have been sent to a virtual destination address have difficulty getting all the way to their target. The term “interior of a network” refers to devices which are not able to directly access another network without first going through another device on their own network. For example, most local area networks (LANs) access the Internet through a proxy server. Devices other than the proxy server are said to be on the interior of the LAN. The proxy server is referred to as an “edge device” because it is able to directly contact another network without using an intermediary device. “Packet filtering” refers to the filtering of incoming messages or packets by an edge device or process on an edge device so that not all of the packets are permitted to proceed to their destination, they are “filtered out”. If the electronic device that is filtering incoming packets, is under the control of the party executing the process associated with the virtual destination address, the device may be configured to allow the packets through to the end destination. However, in many situations, the edge device is not configurable by anyone without system administration privileges. Similarly, if the edge device is a device performing Network Address Translation (a “NAT box”), the NAT box rewrites all outgoing packets from an end user in the interior of the</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>network to make them look like they came directly from the NAT box, and remembers that any traffic coming back from the particular destination address must be mapped back to the originating internal device. Consequently, the responding devices think they are responding to the sending device when they are actually responding to an edge device. In such a case, the packets intended for the virtual destination address on the interior of the physical network may be dropped and not reach their intended destination.” Caronni II at 1:60–2:27.</p> <p>“...sending said formatted message, augmented by said determined MAC address destination, said real IP address destination and said transport protocol header, from a virtual IP address to the real IP address destination indicated in said virtual address destination registration.” Caronni II at 9:20–24.</p> <p>“...a process associated with the sending virtual IP address is located on an electronic device outside a firewall and said virtual address destination is located inside the firewall.” Caronni II at 9:31–32.</p> <p><i>See also</i> Caronni II at 2:31-3:24, 6:32-54, FIGS. 1-5.</p> <p>“The Virtual Network Environment (VNE) of the present invention is defined by a collection of IP addresses. An application running within one VNE can communicate with another application in the same VNE. However, an application in one VNE cannot communicate with an application in another VNE (unless expressly permitted). These and many other attendant advantages of the present invention will be understood upon reading the following detailed description in conjunction with the drawings.” Hipp at 2:8-16.</p> <p>“The VNE is specified at application run time. The VNE is transparent to the application and does not require any modifications to the application. The VNE is defined by subnet of addresses contained within the VNE. For example, all applications within the subnet 10.10.2.0 comprise a VNE. The subnet/netmask specifying such a VNE would be 10.10.2.0/255.255.255.0 and would include the</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>addresses 10.10.2.0 through 10.10.2.255. In this example, an application with IP address 10.10.2.2 would be able to communicate with an application at address 10.10.2.60, but not at 10.10.0.1. Although using a subnet/netmask to specify the VNE is described herein for illustrative purposes, it is to be understood that other methods may be used to accomplish the same mechanism (e.g. an access control list).” Hipp at 3:16-30.</p> <p>“Virtualization of network identity is achieved by assigning a unique virtual IP address and virtual hostname to a group of processes that make up the application instance which the instance keeps throughout its execution. This virtual network identity stays with the application instance regardless of which node the application is running on. The framework, in essence, provides a mechanism to create this virtual network identity (VNI) around the application using the virtual network parameters assigned to it. In one embodiment, the virtual network parameters include an IP address and hostname. The framework 200 ensures that the application's instance uses the virtual network parameters, transparently, so that it can be moved across machines, without modifications to the application.” Hipp at 6:1-15.</p> <p><i>See also</i> Hipp at Figures 1–7.</p> <p>To the extent that Cisco VPN Routers and Software does not disclose any elements of this claim limitation, it would have been obvious to a person having ordinary skill in the art to combine, with the teachings of Cisco VPN Routers and Software, those elements, as known to them or as disclosed by other prior art identified in the cover pleading to Defendant’s Invalidation Contentions.</p>
<p>[30.5] a DNS server for the virtual network, the DNS server configured to receive a DNS request from a first device in the virtual network, and return a network address associated with a</p>	<p>Cisco VPN Routers and Software discloses or renders obvious a DNS server for the virtual network, the DNS server configured to receive a DNS request from a first device in the virtual network, and return a network address associated with a network route director, a private network address associated with a second device in the virtual network, and a virtual network address associated with the second device. For example:</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
<p>network route director, a private network address associated with a second device in the virtual network, and a virtual network address associated with the second device.</p>	<p>“Routing Protocols: RIP, RIP2, OSPF, Static, Automatic endpoint discovery, Network Address Translation (NAT), Classless Interdomain Routing (CIDR).” (Cisco VPN 3000 Series Concentrator Data Sheet).</p> <p>To the extent this reference does not expressly or inherently disclose this limitation, it would have been obvious in view of the background knowledge of a person of ordinary skill in the art. This reference further renders the limitation obvious in light of other references, such as Mehta, RFC 1383, Caronni I, Caronni II, and/or Hipp. Mehta, RFC 1383, Caronni I, Caronni II, and/or Hipp concern similar subject matter as this reference, namely, virtual networks. As set forth in the accompanying contentions, use of a server such as a DNS server that returns network addresses was a well-known and commonly-used technique. Modifying this reference with Mehta’s, RFC 1383’s, Caronni I’s, Caronni II’s, and/or Hipp’s teachings would have only required simple substitution, would not have required extensive experimentation, and would have yielded predictable results. A person of ordinary skill in the art would have further understood how to modify the disclosures of this reference with those of Mehta, RFC 1383, Caronni I, Caronni II, or Hipp for example, to incorporate this limitation. <i>See, e.g.,</i></p> <p>“A network that supports IP can be connected to another TCP/IP-based or UDP/IP-based internet or the Internet, by providing a router which forwards data to another router or host machine based upon an IP address. The router (or routing server/service) typically contains a routing table, which determines to which machine (and optionally to which port) to send a datagram, given a destination IP address. The IP address uniquely identifies a router/host machine, and, in a typical TCP/IP network, can be mapped to a string name that identifies, for example, a particular machine as part of a larger domain. (Although referred to herein often as a TCP/IP-based network, one skilled in the art will recognize that the network may</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>also be UDP-based (connectionless), and may support another session management system.)” Mehta at [0008].</p> <p>“In one embodiment, the AMPS comprises one or more modified DNS/API servers, one or more Address Proxy/Routers, an Address Management Data Server, one or more Address Management Data Repositories, and optionally a load balancer. The AMPS DNS/API server receives a request from a device on a public network for a particular wireless device, and returns an appropriate temporary public address, which is internally mapped to the private address of the wireless device. The public address is then usable by the device on the external public network to send data to the wireless device. The temporary public address is, for example, an address associated with one of the Address Proxy/Routers, which are connected to the external public network and have access to the private addresses on the private wireless network. In some cases, the device on the public network that desires to send data to the wireless device uses a connection-based protocol, such as TCP/IP to send data. In other cases the device uses a connection-less protocol, such as UDP (UDP/IP) to send the data.” Mehta at [0012].</p> <p>“Wireless systems on private networks use techniques similar to Network Address Translation (NAT) technology to send data from a wireless device to the public networking world. In a typical carrier infrastructure that uses a private network, a wireless device “registers” itself with the carrier infrastructure when it powers on (or in other circumstances, when the device attempts to initiate data services). The carrier dynamically assigns the wireless device a private non-routable address by means of DHCP (or DHCP-like) server. The information on the mapping of the transient private IP address to the device public IP address is stored in an internal carrier database and managed by carrier services such as a RADIUS server.” Mehta at [0026].</p> <p>“The Address Management Proxy System achieves two way initiated bi-directional communication by implementing a modified DNS server and serving as a proxy/router for devices on the private wireless network as they interface to the</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>public wired internetworking world. In summary, a pool of public addresses is maintained and dynamically distributed among active wireless devices as needed by the AMPS. FIG. 2 is a block diagram of an example Address Management Proxy System used in bi-directional communication with a wireless device. The term “bi-directional” as used herein means that data paths and communication can flow in either direction between two endpoint systems. FIG. 2 shows wired devices 201, 202, and 203 connected to public network 210. One skilled in the art will recognize that these devices could be connected to another private or public network, which is then connected by one or more wired devices to the public network 210, yet still achieve the functionality discussed here. Any such variation provides equivalent functionality and is explicitly contemplated and presumed to be part of the present invention. On the wireless side, the AMPS 220, acting in its capacity as a proxy (and router) for wireless devices, is shown both connected via wire to the public network 210 and via standard carrier infrastructure elements (not shown) to the wireless network 230. It is presumed that the reader has a working knowledge of the elements of a carrier's infrastructure and the basic mechanisms for routing and mechanisms converting packets from a wired network to a wireless network. These may use analog or digital technology and may require protocol conversions in order to send the physical data and transmit it, for example through a satellite, ultimately to the wireless device. Detailed background information on wireless technology and wireless routing mechanisms is described in Stallings, W., Wireless Communications and Networks, Prentice Hall, N.J., 2002, which is herein incorporated by reference in its entirety. In FIG. 2, wireless device 240 is shown connected via various wireless elements (not shown) to the wireless network 230.” Mehta at [0030].</p> <p>“FIG. 6 is an example block diagram of some of the Address Management Proxy System data repository tables used to support routines of the DNS'/API servers and the Address Proxy/Routers. In one embodiment, the Address Management Data Repository comprises three tables: a unique identifier (unique ID) to private address table 610, a public-to-private address table 630, and a public address to proxy/router machine table 620. Although three tables are shown, one skilled in the</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>art will recognize that these tables could contain other data and may be organized differently, including a different number of tables and with different columns or fields. In addition, any technique for storing a table or list of data may be used. To support embodiments that map a host address plus a port designator to a wireless device, the tables are correspondingly modified.” Mehta at [0052].</p> <p>“The unique ID table 610 maps unique string names of wireless devices to the private wireless network addresses that have been assigned typically by a carrier's infrastructure. In some embodiments, the carrier infrastructure dynamically allocates, using methods similar to a DHCP protocol, a private wireless network address when the wireless device registers itself with the carrier infrastructure upon being powered on. Thus, the unique ID table 610 may be sparsely filled or entries created dynamically and then deleted dynamically as devices register and unregister with the carrier infrastructure system.” Mehta at [0053].</p> <p>The public-to-private address table 630 comprises several fields/columns including a public network address 631, a private (wireless) network address 602, a flag 632 that specifies whether the public address stored in field 631 is free or is already used, a Time to Live (TTL) parameter 633, and other connection data 634. In one embodiment, the DNS'/API servers of the AMPS query table 630 to determine a public network address that corresponds to a designated private wireless network address or to allocate an unused public network address (as indicated in field 632) and map the determined unused public address to a private network address stored in field 602. Mehta at [0054].</p> <p>Public address to proxy/router machine table 620 comprises a public network address field 631 and an indication of a functioning proxy/router machine 621. By maintaining such a mapping, the AMPS is able to substitute proxy/router machines for other proxy/router machines to provide a higher degree of robustness. Each proxy/router machine has a preconfigured set of public network addresses, such as are typically configured by network cards inserted into the proxy/router machine. These address are allocated in a standard fashion through prior purchase or</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>obtaining from an address authorizing authority, currently the Internet Corporation for Assigned Names and Numbers (ICANN). When a machine is inserted for use into the AMPS, indications of these addresses are stored in field 631. Mehta at [0055].</p> <p>“In one embodiment, a timestamp of the mapping between a public network address and a private address is also noted in table 630. After a defined time out, based upon this timestamp, the Address Management Data Server sends a request to the Address Proxy/Router associated with the mapping to unmap the public-to-private address mapping and updates the mapping table 630, thereby returning the associated public address to the pool of unused public network addresses.” Mehta at [0056].</p> <p>“FIG. 7 is an example flow diagram of an example routine provided by a DNS/API server of the Address Management Proxy System to return a public address that corresponds to a designated unique identifier. In essence, this routine implements a DNS query or DNS-like query capability for the AMPS using a modified GetHostByName interface or a specialized API, for example GetProxyIP, as described with reference to FIG. 5. In summary, the routine dynamically allocates an appropriate Address Proxy/Router machine to associate with the wireless device and returns the public address of that machine (along with a TTL parameter and potentially other parameters). Specifically, in step 701, the routine determines the private (non-routable) address of the wireless device designated by a string parameter passed as input to the routine. For example, the string parameter may use fields such as “uniqueID.hostname.domain.tld,” which specifies a typical hierarchy of person/service on a machine named “hostname” on a domain such as a company's network on a top level domain such as “org.” “com,” “edu,” etc. One skilled in the art will recognize that many other string parameter designations could be used. One mechanism for implementing this routine is to request information from the Address Management Data Repository. In one embodiment, the data repository stores a table that maps unique IDs to private network addresses (see Table 610 in FIG. 6). Preferably, any mechanism that is used by the AMPS stores</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>this data in a secure manner in order to keep the wireless network addresses private. In step 702, the routine retrieves the public network address that corresponds to the private wireless network address of the designated device if one has already been assigned by the AMPS to that device and is still valid. In one embodiment, the data repository stores this mapping information between private wireless network address and public network address (see for example table 630 in FIG. 6). If a public network address has not already been assigned or is not valid, then the routine causes a new public network address to be allocated and that new public address is associated with the private wireless network address. Appropriate tables in the data repository are then updated. In step 703, the routine determines the Address Proxy/Router machine that is associated with the assigned public network address (for example using table 620 in FIG. 6). In step 704, the routine sends a request to the determined proxy/router machine to update its routing tables to map the determined public network address to the private wireless network address. In step 705, the routine updates information in the data repository to include any other connection related information (e.g., field 634 in Table 630 in FIG. 6) and indicates a Time to Live (TTL) parameter for the public-private address association (e.g., field 633 in Table 630 in FIG. 6). Once all of the tables have been updated in both the proxy/router and in the data repository, the DNS'/API server returns the determined public network address of the associated Address Proxy/Router machine. As described earlier, the public address may be a (hostname, port) pair when a port-based implementation is used.” Mehta at [0057].</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>The diagram illustrates three data structures and a legend:</p> <ul style="list-style-type: none"> Table 601: A table with two columns: 'Unique ID' (601) and 'Private Network Address' (602). It contains three rows, with the middle row containing an ellipsis (...). Table 621: A table with two columns: 'Public Network Address' (631) and 'Proxy/Router machine' (621). It contains three rows, with the middle row containing an ellipsis (...). Table 630: A larger table with five columns: 'Public Network Address' (631), 'Private Network Address' (602), 'F/U' (632), 'TTL' (633), and 'Other "connection" data' (634). It contains three rows, with the middle row containing an ellipsis (...). Legend: A note stating 'Public/private address → (host address) or (host address, port)'.
	<p><i>Fig. 6</i></p> <p>Mehta at FIG. 6.</p>

Exhibit C-28

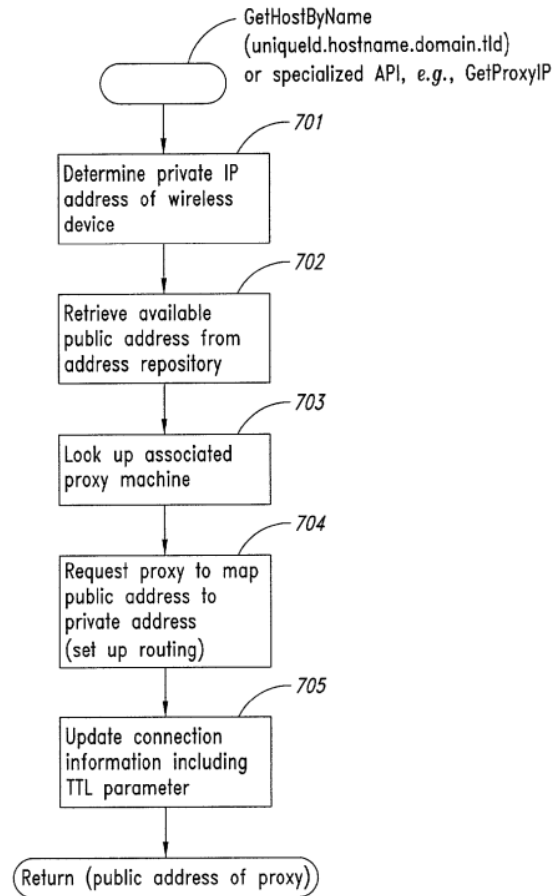


Fig. 7

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>Mehta at FIG. 7.</p> <p><i>See also</i> Mehta at [0004], [0010], [0032], [0036]-[0038], [0048], [0058]-[0059].</p> <p>“1. Routing, scaling and hierarchies</p> <p>Several recent studies have outlined the risk of ‘routing explosion’ in the current Internet: there are already more than 5000 networks announced in the NSFNET routing tables, more than 7000 in the EBONE routing tables. As these numbers are growing, several problems occur:</p> <ul style="list-style-type: none"> * The size of the routing tables grows linearly with the number of connected networks; handling this larger tables requires more resources in all ‘intelligent’ routers, in particular in all ‘transit’ and ‘external’ routers that cannot rely on default routes. * The volume of information carried by the route exchange protocols such as BGP grows with the number of networks, using more network resources and making the reaction to routing events slower. * Explicit administrative decisions have to be exercised by all transit networks administrators which want to implement ‘routing policies’ for each and every additional ‘multi-homed’ network. <p>The current ‘textbook’ solution to the routing explosion problem is to use “hierarchical routing” based on hierarchical addresses. This is largely documented in routing protocols such as IDRP, and is one of the rationales for deploying the CIDR [3] addressing structure in the Internet. This textbook solution, while often perfectly adequate, as a number of inconveniences, particularly in the presence of ‘multihomed stubs’, e.g., customer networks that are connected to more than one service providers.</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>The current proposal presents a scheme that allows for simple routing. It is complementary with the classic ‘hierarchical routing’ approach, but provides an easy to implement and low cost solution for "multi-homed" domains. The solution is a generalization of the ‘MX record’ scheme currently used for mail routing.” RFC 1383 at 1-2.</p> <p>“3.1. Loops and relays</p> <p>In the introduction to DNS-IP routing, we mentioned that the packets would be directed towards the access gateway I1 or I2 by means of ‘source routing’ or ‘tunnelling’. This is not, stricto sensu, necessary. One could imagine that the packet would simply be routed ‘as if it was directed towards I1 or I2’. The next relay would, in turn, also access the DNS to get routing information and forward the packet.</p> <p>Such a strategy would have the advantage of leaving the header untouched and of letting the transit nodes choose the best routing towards the destination, based on their knowledge of the reachability status. It would however have two important disadvantages:</p> <ul style="list-style-type: none">- It would oblige all intermediate relays to access the DNS,- It would oblige all these relays to exploit consistently the DNS information. <p>Obliging all intermediate gateways to access the DNS is impractical in the short term: it would mean that we would have to update each and every transit relay before deploying the scheme. It could also have an important performance impact: the ‘working set’ of transit relays is typical much wider than that of stub gateways, and the argument presented previously on the efficiency of caches may not apply.</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>This would perhaps remain impractical even in the long term, as it the volume of DNS traffic could well become excessive.</p> <p>The second argument would apply even if the performance problem had been solved. Suppose that several RX records are registered for a given destination, such as I1 and I2 for Dx in our example, and that a ‘hop by hop routing’ strategy is used. There would be a fair risk that some relays would choose to route the packet towards I1 and some others towards I2, resulting in inefficient routing and the possibility of loops.</p> <p>In order to ensure coherency, we propose that all routing decisions be made at the source, or by one of the first relays near the source.” RFC 1383 at 4-5.</p> <p>“3.4. Choosing a gateway</p> <p>A simplification to the previous problem would be to allow only one RX record per destination, thus guaranteeing consistent decisions in the network. This would however have a number of draw-backs. A single access point would be a single point of failure, and would be connected to only one transit network thus keeping the ‘customer locking’ effect of hierarchical routing.</p> <p>We propose that the RX records have a structure parallel to that of MX records, i.e., that they carry associated with each gateway address a preference identifier. The source host, when making the routing decision based on RX records, should do the following:</p> <ul style="list-style-type: none"> - List all possible gateways, - Prune all gateways in the list which are known as ‘unreachable’ from the local site,

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<ul style="list-style-type: none">- If the local host is present in the list with a preference index 'x', prune all gateways whose preference index are larger than 'x' or equal to 'x'.- Choose one of the gateway in the list. If the list is empty, consider the destination as unreachable. <p>Indeed, these evaluations should not be repeated for each and every packet. The routers should maintain a cache of the most frequently used destinations, in order to speed up the processing." RFC 1383 at 6.</p> <p>“5.1. DNS record</p> <p>In a definitive scheme, it would be necessary to define a DNS record type and the corresponding 'RX' format. In order to deploy this scheme, we would then have to teach this new format to the domain name service software. While not very difficult to do, this would probably take a couple of month, and will not be used in the early experimentations, which will use the general purpose 'TXT' record.</p> <p>This record is designed for easy general purpose extensions in the DNS, and its content is a text string. The RX record will contain three fields:</p> <ul style="list-style-type: none">- A record identifier composed of the two characters 'RX'. This is used to disambiguate from other experimental uses of the 'TXT' record.- A cost indicator, encoded on up to 3 numerical digits. The corresponding positive integer value should be less that 256, in order to preserve future evolutions.- An IP address, encoded as a text string following the 'dot' notation.

Exhibit C-28

Claims	Cisco VPN Routers and Software								
	<p>The three strings will be separated by a single comma. An example of record would thus be:</p> <table border="1" data-bbox="835 370 1780 480"> <thead> <tr> <th>domain</th> <th>type</th> <th>record</th> <th>value</th> </tr> </thead> <tbody> <tr> <td>*.27.32.192.in-addr.arpa</td> <td>IP</td> <td>TXT</td> <td>RX, 10, 10.0.0.7</td> </tr> </tbody> </table> <p>which means that for all hosts whose IP address starts by the three octets ‘192.32.27’ the IP host ‘10.0.0.7’ can be used as a gateway, and that the preference value is 10.” RFC 1383 at 11-12.</p> <p><i>See also</i> RFC 1383 at 9, 12-13.</p> <p>“To perform this functionality, D 1 108 utilizes a technique known as tunneling to ensure that the communication between itself and enterprise network 102 is secure in that it cannot be viewed by an interloper. ‘Tunneling’ refers to encapsulating one packet inside another when packets are transferred between two end points (e.g., D 1 108 and VPN software 109 running on firewall 106). The packets may be encrypted at their origin and decrypted at their destination. For example, FIG. 2A depicts a packet 200 with a source Internet protocol (IP) address 202, a destination IP address 204, and data 206. It should be appreciated that packet 200 contains other information not depicted, such as the source and destination port. As shown in FIG. 2B, the tunneling technique forms a new packet 208 out of packet 200 by encrypting it and adding both a new source IP address 210 and a new destination IP address 212. In this manner, the contents of the original packet (i.e., 202, 204, and 206) are not visible to any entity other than the destination. Referring back to FIG. 1, by using tunneling, remote device D 1 108 may communicate and utilize the resources of the enterprise network 102 in a secure manner.” Caronni I at 2:26–46.</p> <p>“SASD 540 represents a Supernet: there is one instance of an SASD per Supernet, and it both authenticates nodes and authorizes nodes to join the Supernet. VARPD</p>	domain	type	record	value	*.27.32.192.in-addr.arpa	IP	TXT	RX, 10, 10.0.0.7
domain	type	record	value						
*.27.32.192.in-addr.arpa	IP	TXT	RX, 10, 10.0.0.7						

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>548 has an associated component, VARPDB 551, into which it stores mappings of the internal Supernet addresses, known as a node IDs, to the network addresses recognized by the public-network infrastructure, known as the real addresses. The “node ID” may include the following: a Supernet ID (e.g., 0x123), reflecting a unique identifier of the Supernet, and a virtual address, comprising an IP address (e.g., 10.0.0.1). Although the virtual address is described in an IP address scheme, one skilled in the art will appreciate that the virtual address may be any other type addressing scheme, such as an e-mail address, IPX, or IPv6. Since the node ID includes a Supernet ID, a node will have more than one node ID when it communicates over more than one channel. The ‘real address’ is an IP address (e.g., 10.0.0.2) that is globally unique and meaningful to the public-network infrastructure. In a Supernet, one VARPDB runs on each machine, and it may play two roles. First, a VARPDB may act as a server by storing all address mappings for a particular Supernet into its associated VARPDB. Second, regardless of its role as a server or not, each VARPDB assists in address translation for the nodes on its machine. In this role, the VARPDB stores into its associated VARPDB the address mappings for its nodes, and if it needs a mapping that it does not have, it will contact the VARPDB that acts as the server for the given Supernet to obtain it. The VARPDB may also decide which virtual address to use in the translation. That is, the VARPDB may associate a virtual address with multiple real addresses or vice versa.” Caronni I at 7:3–33.</p> <p>“SNSL 542 utilizes VARPDB 536 to perform address translation. VARPDB stores all of the address mappings encountered thus far by SNSL 542. If SNSL 542 requests a mapping that VARPDB 536 does not have, VARPDB communicates with the VARPDB 526 on the local machine to obtain the mapping. VARPDB 526 will then contact the VARPDB that acts as the server for this particular Supernet to obtain it.” Caronni I at 9:47–54.</p> <p>“FIGS. 7A and 7B depict a flow chart of the steps performed when a node joins a Supernet. The first step performed is that the user invokes the SNlogin script and enters the Supernet name, their user ID, their password, and a requested virtual</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>address (step 702). Of course, this information depends on the particular authentication mechanism used. Upon receiving this information, the SNlogin script performs a handshaking with SASD to authenticate this information. In this step, the user may request a particular virtual address to be used, or alternatively, the SASD may select one for them. Next, if any of the information in step 702 is not validated by SASD (step 704), processing ends. Otherwise, upon successful authentication, SASD creates an address mapping between a node ID and the real address (step 706). In this step, SASD concatenates the Supernet ID with the virtual address to create the node ID, obtains the real address of the SNlogin script by querying network services in a well-known manner, and then registers this information with the VARPDB that acts as the server for this Supernet. This VARPDB is identified in the configuration file.” Caronni I at 9:66-10:18.</p> <p>“The packet and Supernet ID are then transmitted to the SNSL layer using the modified socket structure (step 806). The SNSL layer then accesses the VARPDB to obtain the address mapping between virtual source node address 642 and the source real address 614 as well as the virtual destination node address 644 and the destination real address 616 (step 808). If they are not contained in the VARPDB because this is the first time a packet has been sent from this node or sent to this destination, the VARPDB accesses the local VARPDB to obtain the mapping. When contacted, the VARPDB on the local machine contacts the VARPDB that acts as the server for the Supernet to obtain the appropriate address mapping. Since the VARPDB maintains all real IP addresses, a remote node may securely communicate with another remote node without reverfication.” Caronni I at 11:47–61.</p> <p>“After obtaining the address mapping, the SNSL layer determines whether it has been configured to communicate over the appropriate channel for this packet (step 806). This configuration occurs when SNlogin runs, and if the SNSL has not been so configured, processing ends. Otherwise, SNSL obtains the channel key to be used for this channel (step 808). The SNSL maintains a local cache of keys and an indication of the channel to which each key is associated. Each channel key is time</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>stamped to expire in ten seconds, although this time is configurable by the administrator. If there is a key located in the cache for this channel, SNSL obtains the key. Otherwise, SNSL accesses KMD which then locates the appropriate channel key from the appropriate KMC. After obtaining the key, the SNSL layer encrypts the packet using the appropriate encryption algorithm and the key previously obtained (step 810). When encrypting the packet, the virtual source node address 642, the virtual destination node address 644, and the data may be encrypted (addressing section 660), but the source and destination real addresses 614, 616 (delivery scheme section 670) are not, so that the real addresses can be used by the public network infrastructure to send the packet to its destination. By encrypting addressing scheme 660, the Supernet enables data to be transmitted securely and at the same time transparent from delivery scheme used.” Caronni I at 11:62-12:19.</p> <p>“Web client 1102 has a virtual address obtained from computer system 1106, as described in FIGS. 7A and 7B. Each time web client 1102 requests a packet from web server 1104 a, the client requests the virtual address of the web server 1104 a from computer system 1106. If web server 1104 a becomes overloaded (e.g., unable to handle more requests), the overloaded web server spawns new instances of the same web server (web servers 1104 b and 1104 c) and at the same time notifies the computer system 1106. The VARP server of computer 1106 then associates the new instances 104 b and 104 c of the web server with web server's 1104 virtual address. As a result, web client 1102 is not notified of the change and continues to use the same virtual address as previously.” Caronni I at 13:13–26.</p> <p>“In one embodiment, a virtual network supported by a physical network includes a virtual address resolution facility. The virtual address resolution facility is used to resolve a virtual IP address into a real IP address. A virtual address is registered with the virtual address resolution facility and the registration includes a real IP address, a port number and a transport protocol designation associated with a virtual address. The virtual address destination is resolved using the virtual address resolution facility and a message is sent from a virtual address in the virtual</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>network to the real IP address indicated in the virtual address destination registration. In one aspect of the embodiment, the registration also includes an Application layer protocol designation.” Caronni II at 2:52–64.</p> <p>“In an embodiment of the present invention a virtual network is supported by a physical network and includes a virtual address resolution facility. The virtual address resolution facility is used to resolve a virtual IP address into a real IP address. A registration request is received at the virtual address resolution facility which includes a real IP address, a port address, a transport protocol designation and an Application layer protocol designation associated with a virtual address destination. The associations are stored using the virtual address resolution facility. A resolution request is received referencing a virtual address destination and the resolution request is resolved using the virtual address resolution facility and the stored associations.” Caronni II at 3:11–23.</p> <p>“The physical network 20 is also interfaced with an electronic device 30 located at the edge of a local area physical network 50 which includes devices addressable using IP addresses in the 129.63.1.0/24 range. The electronic device 30 may be a proxy server, gateway, NAT box, or other device and may perform packet filtering. Alternatively, a firewall 51 may be executed in either a hardware or software form to filter packets sent to the local area physical network 50 from the physical network 20. The firewall 51 may be located outside the electronic device 30 or alternatively may be running in software form on the electronic device 30. Those skilled in the art will recognize that a number of different types of networks may be utilized within the scope of the present invention. The networks may be the Internet, an intranet, wide area network (WAN), a local area network (LAN), a satellite network, a wireless network, or some other type of network capable of supporting a virtual network through the devices thereon.” Caronni II at 4:17–33.</p> <p>“FIG. 2 depicts a block diagram of the VARP lookup table 26 used by the virtual address resolution facility 24. The VARP lookup table 26 may be stored at any location accessible over the network. The VARP lookup table 26 includes</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>registered virtual destination addresses located in a column 78 and corresponding associations located in a column 79. Entries 80, 82, 84, 86 and 88 include the registered virtual destination address and information associated with the registered virtual destination address. Each entry in the VARP lookup table 26 will list the registering virtual IP address 90 and associated information including a real IP address 91, a transport protocol designation 92, a port number identification 93 and an Application layer protocol designation 94. The associations are provided at the time the virtual destination address is registered in the virtual network. For example, the entry 80 for the virtual IP address 10.0.0.12 (90) includes the association of a real IP address 152.70.0.1 (91), an associated Transport layer protocol designation, UDP 92, an associated port number, 6789 (93), and an Application layer protocol designation of ‘none’ 94.” Caronni II at 5:10–29.</p> <p>“The illustrative embodiment of the present invention attempts to circumvent problems caused by packet filtering and network address translation occurring at the edge of networks through the use of higher layer protocols. FIG. 3 depicts a block diagram of the encapsulation process used to send a message from a virtual IP address to another virtual IP address located at the interior of a network at which packet filtering and/or network address translation is occurring. The virtual address resolution facility 24 is used to determine the associations registered with the destination address. The original message 111 at the Application layer 110 is formatted in a format specified in the registration of the virtual destination address if one is so indicated (e.g. HTTP). The message 111 is passed down to the Transport layer 112. The Transport layer 112 takes the original message 111 and adds a transport protocol header 113 as specified in the virtual address destination registration. The transport protocol header 113 may be a UDP header, TCP header, an X.25 header, XTP header, AppleTalk header, or other similar transport protocol header. Those skilled in the art will realize that the message may include an IPSEC header, inner IP header, UDP or TCP header and inner payload.” Caronni II at 6:32-53.</p> <p>“The illustrative embodiment of the present invention may also be utilized to send a</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>message from a virtual address located behind a NAT box to a destination address on the same virtual network that is behind a different NAT box. Ordinarily, where the destination address is directly connected to the Internet (i.e.: in situations where the destination is not behind a NAT box), a connection, such as a TCP connection, may be established between the originator of the connection behind the NAT box and the destination. The originator's message passes the NAT box and the internal mapping of public address and port number to internal address and port number takes place. Return packets from the destination may then be received following the mapping. When the destination address is also behind a NAT box, the originator of the message is unable to directly address the destination (since the address of the destination may not be routable in the public Internet). In such a case, the present invention adds an entry to the VARP table for a third-party reflecting agent located at an address outside the NAT box. Messages that are being sent from the originating virtual address behind a NAT box to a destination which is behind a different NAT box are sent via the reflecting agent intermediary outside the NAT box and reflected to the destination.</p> <p>FIG. 5 depicts a block diagram of an environment holding a virtual network in which both the originator and destination of a message are located behind NAT boxes. An electronic device holding an originating process 182 which is part of a virtual network is interfaced with a network 180, such as the Internet, via a NAT box 186. An intended recipient for a message sent from the originating process 180 is an electronic device with a destination address 184 that is also behind a NAT box 188. A reflecting agent 200, the address of which has been added to a VARP table, acts as an intermediary to allow a connection between the originating process 182 and destination address 184.” Caronni II at 8:1–38.</p>

Exhibit C-28

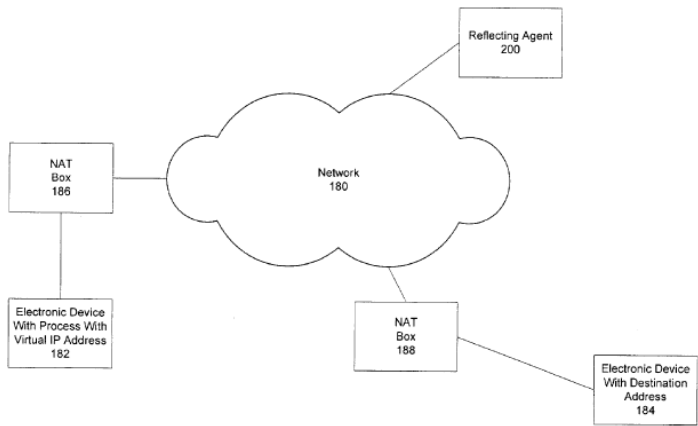
Claims	Cisco VPN Routers and Software
	<p style="text-align: right;">Figure 5</p>  <p>Caronni II at FIG. 5.</p> <p>“Beginning at step 270, a client application requests to the VNE framework 200 to connect or send to an address of another application, such as a server application having the address 10.10.2.70: port 9001. At step 272, the VNE framework 200 requests the VNE parameters for the process corresponding to the client application from a process state storage structure. The structure that stores process state is a structure that the operating system uses to store private information about the process. Therefore, it may differ depending on the operating system. The parameters added the process state storage structure as part of the virtual network environment are listed below:</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<hr/> <pre> typedef struct { ipaddr_t app_address; ipaddr_t virtual_subnet; ipaddr_t virtual_mask; ipaddr_t global_subnet; ipaddr_t global_mask; } vne_param_t; </pre> <hr/> <p>Hipp at 4:44–64.</p> <p>“Virtualization of network identity is achieved by assigning a unique virtual IP address and virtual hostname to a group of processes that make up the application instance which the instance keeps throughout its execution. This virtual network identity stays with the application instance regardless of which node the application is running on. The framework, in essence, provides a mechanism to create this virtual network identity (VNI) around the application using the virtual network parameters assigned to it. In one embodiment, the virtual network parameters include an IP address and hostname. The framework 200 ensures that the application's instance uses the virtual network parameters, transparently, so that it can be moved across machines, without modifications to the application.</p> <p>The virtual hostname resolves to the virtual IP address for both the applications registered with the VNI framework as well as those that are not registered. This may require configuration of a name service or OS host configuration files. For example, if an application instance used a virtual IP address of 10.10.0.1 and a virtual hostname of host1055, the standard hostname to IP address resolution mechanisms (e.g. DNS or the /etc/hosts file) would have to be preconfigured to resolve a query of host1055 to IP address 10.10.0.1.” Hipp at 6:1–26.</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>To the extent that Cisco VPN Routers and Software does not disclose any elements of this claim limitation, it would have been obvious to a person having ordinary skill in the art to combine, with the teachings of Cisco VPN Routers and Software, those elements, as known to them or as disclosed by other prior art identified in the cover pleading to Defendant's Invalidity Contentions.</p>
<p>35. The virtual network manager of claim 30 further comprising a join module configured to receive a join request from the agent associated with the device to indicate that the device is connected for data communication within the virtual network, the join module further configured to receive a leave request from the agent associated with the device to indicate that the device will be disconnected from data communication within the virtual network.</p>	<p>Cisco VPN Routers and Software discloses or renders obvious the virtual network manager of claim 30 further comprising a join module configured to receive a join request from the agent associated with the device to indicate that the device is connected for data communication within the virtual network, the join module further configured to receive a leave request from the agent associated with the device to indicate that the device will be disconnected from data communication within the virtual network. For example:</p> <p style="padding-left: 40px;">“The client can be pre-configured for mass deployments and the initial logons require very little user intervention.” (Cisco VPN 3000 Series Concentrator Data Sheet).</p> <p>To the extent this reference does not expressly or inherently disclose this limitation, it would have been obvious in view of the background knowledge of a person of ordinary skill in the art. This reference further renders the limitation obvious in light of other references, such as Mehta and/or Caronni I. Mehta and Caronni I concern similar subject matter as this reference, namely, virtual networks. As set forth in the accompanying contentions, joining devices with a virtual network was a well-known and commonly-used technique. Modifying this reference with Mehta's and/or Caronni I's teachings would have only required simple substitution, would not have required extensive experimentation, and would have yielded predictable results. A person of ordinary skill in the art would have further understood how to modify the disclosures of this reference with that of Mehta and Caronni I for example, to incorporate this limitation. <i>See, e.g.,</i></p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>“Wireless systems on private networks use techniques similar to Network Address Translation (NAT) technology to send data from a wireless device to the public networking world. In a typical carrier infrastructure that uses a private network, a wireless device “registers” itself with the carrier infrastructure when it powers on (or in other circumstances, when the device attempts to initiate data services). The carrier dynamically assigns the wireless device a private non-routable address by means of DHCP (or DHCP-like) server. The information on the mapping of the transient private IP address to the device public IP address is stored in an internal carrier database and managed by carrier services such as a RADIUS server.” Mehta at [0026].</p> <p>“FIG. 4 is an example block diagram of a general purpose computer system for practicing embodiments of the Address Management Proxy System. The general purpose computer system 400 may comprise one or more server and/or client computing systems and may span distributed locations. In addition, each block may represent one or more such blocks as appropriate to a specific embodiment or may be combined with other blocks. The various blocks of the Address Management Proxy System 410 may physically reside on one or more machines, which use standard interprocess communication mechanisms to communicate with each other. In the embodiment shown, computer system 400 comprises a computer memory (“memory”) 01, a display 402, a Central Processing Unit (“CPU”) 403, and Input/Output devices 404. The Address Management Proxy System 410 is shown residing in memory 401. The components of the Address Management Proxy System 410 preferably execute on CPU 403 and manage the address mapping of wireless devices on a wireless network, as described in previous figures, to allow other wired systems to communicate with the wireless devices. Other downloaded code 405 and potentially other data repositories also reside in the memory 410, and preferably execute on one or more CPU's 403. In a typical embodiment, the AMPS 410 includes one or more DNS/API servers 411, one or more Address Proxy/Routers 412, an Address Management Data Server 413, and Address Management Data Repositories 414. As described earlier, the AMPS may include</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>other data repositories and components, such as a load balancer, depending upon the particular implementation.” Mehta at [0036].</p> <p>“The unique ID table 610 maps unique string names of wireless devices to the private wireless network addresses that have been assigned typically by a carrier's infrastructure. In some embodiments, the carrier infrastructure dynamically allocates, using methods similar to a DHCP protocol, a private wireless network address when the wireless device registers itself with the carrier infrastructure upon being powered on. Thus, the unique ID table 610 may be sparsely filled or entries created dynamically and then deleted dynamically as devices register and unregister with the carrier infrastructure system.” Mehta at [0053].</p> <p>“FIGS. 7A and 7B depict a flow chart of the steps performed when a node joins a Supernet. The first step performed is that the user invokes the SNlogin script and enters the Supernet name, their user ID, their password, and a requested virtual address (step 702). Of course, this information depends on the particular authentication mechanism used. Upon receiving this information, the SNlogin script performs a handshaking with SASD to authenticate this information. In this step, the user may request a particular virtual address to be used, or alternatively, the SASD may select one for them. Next, if any of the information in step 702 is not validated by SASD (step 704), processing ends. Otherwise, upon successful authentication, SASD creates an address mapping between a node ID and the real address (step 706). In this step, SASD concatenates the Supernet ID with the virtual address to create the node ID, obtains the real address of the SNlogin script by querying network services in a well-known manner, and then registers this information with the VARPd that acts as the server for this Supernet. This VARPd is identified in the configuration file.” Caronni I at 9:67–10:18.</p> <p>“FIG. 10 depicts a flow chart of the steps performed when logging a node out of a Supernet. The first step performed is for the user to run the SNlogout script and to</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>enter a node ID (step 1002). Next, the SNlogout script requests a log out from SASD (step 1004). Upon receiving this request, SASD removes the mapping for this node from the VARPd that acts as the server for the Supernet (step 1006). SASD then informs KMS to cancel the registration of the node, and KMS terminates this KMC (step 1008). Lastly, KMS generates a new channel key for the channels on which the node was communicating (step 1010) to reduce the likelihood of an intruder being able to intercept traffic.” Caronni I at 12:61–13:5.</p> <p><i>See also</i> Caronni I at FIG. 10.</p> <p>To the extent that Cisco VPN Routers and Software does not disclose any elements of this claim limitation, it would have been obvious to a person having ordinary skill in the art to combine, with the teachings of Cisco VPN Routers and Software, those elements, as known to them or as disclosed by other prior art identified in the cover pleading to Defendant’s Invalidity Contentions.</p>
<p>37. The virtual network manager of claim 35 wherein the join module is further configured to maintain data to associate a virtual network address with a device in the virtual network.</p>	<p>Cisco VPN Routers and Software discloses or renders obvious the virtual network manager of claim 35 wherein the join module is further configured to maintain data to associate a virtual network address with a device in the virtual network. For example:</p> <p>“VPN access policies are created and stored centrally in the Cisco VPN 3000 Concentrator and pushed to the client when a connection is established.” (Cisco VPN 3000 Series Concentrator Data Sheet).</p> <p>“The VPN encryption modules handle a variety of IPSec-related tasks, including encryption, hashing, key exchange, storage of security associations.” (Cisco 2600 and 3600 VPN Router Bundles).</p> <p>To the extent this reference does not expressly or inherently disclose this limitation, it would have been obvious in view of the background knowledge of a person of ordinary skill in the art. This reference further renders the limitation obvious in light of other</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>references, such as Mehta and/or Caronni I. Mehta and Caronni I concern similar subject matter as this reference, namely, virtual networks. As set forth in the accompanying contentions, joining devices with a virtual network was a well-known and commonly-used technique. Modifying this reference with Mehta’s and/or Caronni I’s teachings would have only required simple substitution, would not have required extensive experimentation, and would have yielded predictable results. A person of ordinary skill in the art would have further understood how to modify the disclosures of this reference with that of Mehta and Caronni I for example, to incorporate this limitation. <i>See, e.g.,</i></p> <p>“In one embodiment, the AMPS comprises one or more modified DNS/API servers, one or more Address Proxy/Routers, an Address Management Data Server, one or more Address Management Data Repositories, and optionally a load balancer. The AMPS DNS/API server receives a request from a device on a public network for a particular wireless device, and returns an appropriate temporary public address, which is internally mapped to the private address of the wireless device. The public address is then usable by the device on the external public network to send data to the wireless device. The temporary public address is, for example, an address associated with one of the Address Proxy/Routers, which are connected to the external public network and have access to the private addresses on the private wireless network. In some cases, the device on the public network that desires to send data to the wireless device uses a connection-based protocol, such as TCP/IP to send data. In other cases the device uses a connection-less protocol, such as UDP (UDP/IP) to send the data.” Mehta at [0012].</p> <p>“FIG. 3 is an example block diagram of components of an example Address Management Proxy System. In one embodiment, the Address Management Proxy System (AMPS) comprises one or more modified DNS/ API servers 302, one or more Address Proxy/Routers 305, an Address Management Data Server 303 which manages a database or other repositories such as Address Management Data Repository 304, and optionally a load balancer 301. The DNS/API servers 302 are either individually connected to a public network 310 or are connected to the load</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>balancer 301 which is in turn connected to public network 310. Similarly, each Address Proxy/Router 305 is also connected to the public network 310, via the routable (public) address to which data from the external network 310 is sent that is destined for the wireless devices. The DNS'/API servers 302 are either modified implementations of a DNS server to add functionality necessary to communicate with wireless devices, or are servers that implement one or more specialized APIs, as will be described further below. The DNS'/API servers 302 use the Address Management Data Server 303 to assist in mapping a unique identifier (e.g., string name) for a wireless device to a public address on public network 310. The pool of public addresses is also maintained by the Address Management Data Serve 303 and Data Repository 304. The Address Management Data server 303 and Address Management Data Repository 304 may be implemented using existing database technology, for example, ODBC technology or, may be implemented as a structure such as a simple text file. One skilled in the art will recognize that any embodiment for storing a set of tables, data, lists, or mappings can be used. Each Address Proxy/Router 305 also uses the Address Management Data Server 303 or equivalent to create and update a series of routing tables that are used to assign public addresses to wireless devices as needed and to update the various mappings between public addresses and the non-routable (private) addresses of the wireless devices. The tables and mappings that are maintained on behalf of the DNS'/API servers 302 and the Address Proxy/Routers 305 by the Address Management Data Server 303 are described below with reference to FIG. 6.” Mehta at [0032].</p> <p>“One skilled in the art will recognize that the AMPS 410 may be implemented in a distributed environment that is comprised of multiple, even heterogeneous, computer systems and networks. For example, in one embodiment, the DNS'/API servers 411, the Address Proxy/Router components 412, and the Address Management Data Servers 413 with their data repositories 414 are all located in physically different computer systems. In another embodiment, various components of the AMPS 410 are hosted each on a separate server machine and may be remotely located from the tables which are stored in the address management data repository 414. In addition, under some scenarios, the entire AMPS system 410 may</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>be hosted within a carrier's infrastructure and be completely subsumed by it. Different configurations and locations of programs and data are contemplated for use with techniques of the present invention. In example embodiments, these components may execute concurrently and asynchronously; thus the components may communicate using well-known message passing techniques. One skilled in the art will recognize that equivalent synchronous embodiments are also supported by an AMPS implementation. Also, other steps could be implemented for each routine, and in different orders, and in different routines, yet still achieve the functions of the AMPS.” Mehta at [0038].</p> <p>“FIG. 7 is an example flow diagram of an example routine provided by a DNS/API server of the Address Management Proxy System to return a public address that corresponds to a designated unique identifier. In essence, this routine implements a DNS query or DNS-like query capability for the AMPS using a modified GetHostByName interface or a specialized API, for example GetProxyIP, as described with reference to FIG. 5. In summary, the routine dynamically allocates an appropriate Address Proxy/Router machine to associate with the wireless device and returns the public address of that machine (along with a TTL parameter and potentially other parameters). Specifically, in step 701, the routine determines the private (non-routable) address of the wireless device designated by a string parameter passed as input to the routine. For example, the string parameter may use fields such as “uniqueID.hostname.domain.tld,” which specifies a typical hierarchy of person/service on a machine named “hostname” on a domain such as a company's network on a top level domain such as “org.” “com,” “edu,” etc. One skilled in the art will recognize that many other string parameter designations could be used. One mechanism for implementing this routine is to request information from the Address Management Data Repository. In one embodiment, the data repository stores a table that maps unique IDs to private network addresses (see Table 610 in FIG. 6). Preferably, any mechanism that is used by the AMPS stores this data in a secure manner in order to keep the wireless network addresses private. In step 702, the routine retrieves the public network address that corresponds to the private wireless network address of the designated device if one has already been</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>assigned by the AMPS to that device and is still valid. In one embodiment, the data repository stores this mapping information between private wireless network address and public network address (see for example table 630 in FIG. 6). If a public network address has not already been assigned or is not valid, then the routine causes a new public network address to be allocated and that new public address is associated with the private wireless network address. Appropriate tables in the data repository are then updated. In step 703, the routine determines the Address Proxy/Router machine that is associated with the assigned public network address (for example using table 620 in FIG. 6). In step 704, the routine sends a request to the determined proxy/router machine to update its routing tables to map the determined public network address to the private wireless network address. In step 705, the routine updates information in the data repository to include any other connection related information (e.g., field 634 in Table 630 in FIG. 6) and indicates a Time to Live (TTL) parameter for the public-private address association (e.g., field 633 in Table 630 in FIG. 6). Once all of the tables have been updated in both the proxy/router and in the data repository, the DNS/API server returns the determined public network address of the associated Address Proxy/Router machine. As described earlier, the public address may be a (hostname, port) pair when a port-based implementation is used.” Mehta at [0057].</p> <p>“Specifically, in step 801, the routine determines (for example, from the Address Management Data Repository) the private wireless address that corresponds to the invoked public address and the TTL parameter associated with this mapping. These values can be obtained, for example, from the public-to-private address table 630 of FIG. 6. In step 802, the routine determines whether the value of the determined TTL parameter indicates that the mapping has expired, and if so, returns an error, else continues in step 803. In step 803, the routine determines the format required by the target device (the wireless network format). In step 804, the routine determines whether any protocol conversion is necessary and, if so, continues in step 805 else continues in step 806. Note that protocol conversion for a specific wireless network such as converting the data to an “HTTP” packet may be done by the proxy/router or may be done by some other component within the carrier infrastructure. One</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>skilled in the art will recognize that these are example steps and that different formatting or different protocol conversion routines may be added or omitted as specific to the environment. In step 806, the Address Proxy/Router routine sends the data packet (which has been formatted and whose protocol is converted as necessary) to the determined associated private address of the wireless device, and returns.” Mehta at [0059].</p> <div style="text-align: center;"> </div> <p align="center"><i>Fig. 6</i></p> <p>Mehta at FIG. 6.</p> <p>“SASD 540 represents a Supernet: there is one instance of an SASD per Supernet, and it both authenticates nodes and authorizes nodes to join the Supernet. VARPD</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>548 has an associated component, VARPDB 551, into which it stores mappings of the internal Supernet addresses, known as a node IDs, to the network addresses recognized by the public-network infrastructure, known as the real addresses. The “node ID” may include the following: a Supernet ID (e.g., 0x123), reflecting a unique identifier of the Supernet, and a virtual address, comprising an IP address (e.g., 10.0.0.1). Although the virtual address is described in an IP address scheme, one skilled in the art will appreciate that the virtual address may be any other type addressing scheme, such as an e-mail address, IPX, or IPv6. Since the node ID includes a Supernet ID, a node will have more than one node ID when it communicates over more than one channel. The “real address” is an IP address (e.g., 10.0.0.2) that is globally unique and meaningful to the public-network infrastructure. In a Supernet, one VARPDB runs on each machine, and it may play two roles. First, a VARPDB may act as a server by storing all address mappings for a particular Supernet into its associated VARPDB. Second, regardless of its role as a server or not, each VARPDB assists in address translation for the nodes on its machine. In this role, the VARPDB stores into its associated VARPDB the address mappings for its nodes, and if it needs a mapping that it does not have, it will contact the VARPDB that acts as the server for the given Supernet to obtain it. The VARPDB may also decide which virtual address to use in the translation. That is, the VARPDB may associate a virtual address with multiple real addresses or vice versa.” Caronni I at 7:3–34.</p> <p>“FIGS. 7A and 7B depict a flow chart of the steps performed when a node joins a Supernet. The first step performed is that the user invokes the SNlogin script and enters the Supernet name, their user ID, their password, and a requested virtual address (step 702). Of course, this information depends on the particular authentication mechanism used. Upon receiving this information, the SNlogin script performs a handshaking with SASD to authenticate this information. In this step, the user may request a particular virtual address to be used, or alternatively, the SASD may select one for them. Next, if any of the information in step 702 is not validated by SASD (step 704), processing ends. Otherwise, upon successful authentication, SASD creates an address mapping between a node ID and the real address (step</p>

Exhibit C-28

Claims	Cisco VPN Routers and Software
	<p>706). In this step, SASD concatenates the Supernet ID with the virtual address to create the node ID, obtains the real address of the SNlogin script by querying network services in a well-known manner, and then registers this information with the VARPd that acts as the server for this Supernet. This VARPd is identified in the configuration file.” Caronni I at 9:67–10:18.</p> <p>“FIG. 10 depicts a flow chart of the steps performed when logging a node out of a Supernet. The first step performed is for the user to run the SNlogout script and to enter a node ID (step 1002). Next, the SNlogout script requests a log out from SASD (step 1004). Upon receiving this request, SASD removes the mapping for this node from the VARPd that acts as the server for the Supernet (step 1006). SASD then informs KMS to cancel the registration of the node, and KMS terminates this KMC (step 1008). Lastly, KMS generates a new channel key for the channels on which the node was communicating (step 1010) to reduce the likelihood of an intruder being able to intercept traffic.” Caronni I at 12:61–13:5.</p> <p><i>See also</i> Caronni I at FIG. 10.</p> <p>To the extent that Cisco VPN Routers and Software does not disclose any elements of this claim limitation, it would have been obvious to a person having ordinary skill in the art to combine, with the teachings of Cisco VPN Routers and Software, those elements, as known to them or as disclosed by other prior art identified in the cover pleading to Defendant’s Invalidity Contentions.</p>

Exhibit C-29

Invalidity of U.S. Patent No. 7,949,785 in View of VMware Systems

VMware Systems was known in this country, used in public, sold or offered for sale no later than 2002, as demonstrated by at least the materials cited herein or the testimony of knowledgeable witnesses and corroborating documents.¹ Specifically, based on information currently available, VMware Systems was (1) known or used in this country before the alleged invention of the claimed subject matter of the asserted claims, (2) in public use or on sale in this country more than one year before the filing date of the patent, or (3) invented by another who did not abandon, suppress, or conceal, before the alleged invention of the claimed subject matter of the asserted claims. Thus, VMware Systems anticipates the asserted claims under at least 35 U.S.C. §§ 102(a) or (g) as to claims 30, 35, and 37 (the “Asserted Claims”) of the ’785 Patent, as discussed in detail in the chart that follows.

At least the following documents describe the relevant functionality disclosed by VMware Systems:

- VMware GSX Server 3.1 Administration Guide;
- VMware Delivers Open Interface to Virtual Infrastructure;
- VMware Enterprises Continue to Embrace VMware Virtual Infrastructure;
- VMware ESX Server Specifications;
- VMware GSX Server Specifications;
- VMware GSX Server 3.1 Virtual Machine Guide; and
- VMware Scripting API User’s Manual.

Based on information currently available, the persons or entities involved in the conception and diligent reduction to practice of the ideas in VMware Systems include at least persons involved in the VMware Systems and their testing and commercial use, including at least VMware Systems. The conception of VMware Systems occurred at least as early as 2002.

Defendant offers this invalidity chart based on its current understanding of VMware Systems. Defendant’s investigation of this system is ongoing. Defendant may rely on materials produced by Plaintiffs or by third parties regarding VMware Systems and related software to demonstrate invalidity of the asserted claims. Defendant has not had an opportunity to review source code for

¹ Defendant’s invalidity charts, in some instances, rely at least in part on Plaintiff’s apparent positions regarding the scope of its claims for purposes of asserting infringement. Nothing in these claim charts should be understood as an admission relating to infringement, either literally or under the doctrine of equivalents, or as an admission relating to Defendant’s understanding of the proper interpretation or scope of the Asserted Claims. Defendant reserves the right to rely on additional citations or sources of evidence that may also be applicable, or that may become applicable in light of any Court Order on claim construction, changes in Plaintiff’s infringement contentions, or information obtained during discovery as the case proceeds.

Exhibit C-29

VMware Systems. Defendant may rely on source code, when available, implementing the functionality described herein, or other source code implementing substantially similar functionality, to demonstrate the invalidity of the asserted claims.

To the extent Plaintiff alleges VMware Systems does not disclose any particular limitation of the Asserted Claims of the '785 Patent, either expressly or inherently, any purported differences are such that the claimed subject matter as a whole would have been obvious in view of the knowledge of one skilled in the art. It would have further been obvious to a person of ordinary skill in the art as of the priority date of the '785 Patent to modify VMware Systems or combine the teachings of VMware Systems with other prior art in a manner that would have rendered the Asserted Claims invalid as obvious, including but not limited to:

- U.S. Patent Application Publication No. 2003/0028671 (“Mehta”);
- Huitema, C., Network Working Group Request for Comment (RFC): 1383, entitled An Experiment in DNS Based IP Routing (“RFC 1383”);
- U.S. Patent No. 6,970,941 (“Caronni I”);
- U.S. Patent No. 7,814,228 (“Caronni II”); and/or
- U.S. Patent No. 6,766,371 (“Hipp”).

As the United States Supreme Court held in *KSR Int'l Co. v. Teleflex, Inc.*, “[t]he combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results.” 550 U.S. 398, 416 (2007). The Supreme Court further held: “[w]hen a work is available in one field of endeavor, design incentives and other market forces can prompt variations of it, either in the same field or a different one. If a person of ordinary skill can implement a predictable variation, § 103 likely bars its patentability. For the same reason, if a technique has been used to improve one device, and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond his or her skill.” *Id.* at 417. The Supreme Court further held: “[w]hen there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, a person of ordinary skill has good reason to pursue the known options within his or her technical grasp. If this leads to the anticipated success, it is likely the product not of innovation but of ordinary skill and common sense.” *Id.* at 421.

Defendant reserves the right to amend or supplement this claim chart at a later date as more fully set forth in the Invalidity Contentions.

Exhibit C-29

Claims	VMware Systems
<p>[30.pre] A virtual network manager, comprising:</p>	<p>To the extent the preamble is limiting, VMware Systems discloses or renders obvious a virtual network manager. For example:</p> <p>“VMware® GSX Server™ is virtual infrastructure for enterprise IT administrators who want to consolidate servers and streamline development and testing operations.” (VMware GSX Server 3.1 Administration Guide)</p> <p>“VMware, Inc., the global leader in virtual infrastructure software for x86-based systems, today announced the availability of the VMware Virtual Infrastructure SDK. The SDK provides standards-based interfaces to control VMware virtual infrastructure.” (VMware Delivers Open Interface to Virtual Infrastructure)</p> <p>“Virtual infrastructure provides a layer of abstraction between the computing, storage and networking hardware and the software that runs on it.” (VMware Enterprises Continue to Embrace VMware Virtual Infrastructure)</p> <p>To the extent this reference does not expressly or inherently disclose this limitation, it would have been obvious in view of the background knowledge of a person of ordinary skill in the art. This reference further renders the limitation obvious in light of other references, such as Mehta and/or Caronni I. Mehta and Caronni I concern similar subject matter as this reference, namely, virtual networks. As set forth in the accompanying contentions, use of virtual networks was well-known and common. Modifying this reference with Mehta’s and Caronni I’s teachings would have only required simple substitution, would not have required extensive experimentation, and would have yielded predictable results. A person of ordinary skill in the art would have further understood how to modify the disclosures of this reference with that of Mehta and Caronni I for example, to incorporate this limitation. <i>See, e.g.,</i></p> <p>“Methods and systems for providing two-way initiated, bi-directional communication with wireless devices using connection-based or connection-less</p>

Exhibit C-29

Claims	VMware Systems
	<p>protocols, such as, for example, TCP/IP and UDP/IP, are provided. Example embodiments provide an Address Management Proxy System (“AMPS”), which enables devices and systems connected to a public internet, such as the Internet, to initiate communication with and to send data to wireless devices connected to a private wireless network, without exposing the non-routable private addresses of these wireless devices. The AMPS allocates a public (routable) network address for temporarily use by a requesting device on a public network to communicate with a wireless device on a wireless network. In one embodiment, a pool of public addresses, for example, public IP addresses, is maintained by the AMPS and allocated dynamically to wireless network devices as required. In one embodiment, the AMPS comprises one or more modified DNS/API servers, one or more Address Proxy/Routers, an Address Management Data Server, one or more data repositories, and optionally a load balancer. The AMPS DNS/API server receives a request from a device on a public network for a particular wireless device, and returns an appropriate temporary public address, which is internally mapped to the private address of the wireless device. The public address is then usable by the device on the public network to send data to the wireless device.” Mehta at Abstract.</p> <p>“The present invention relates to methods and systems for initiating communication with wireless devices and, in particular, to methods and systems for initiating communication with a device on a private network from a device on a public network to achieve virtual end-to-end connectivity.” Mehta at [0002].</p> <p>“Embodiments of the present invention provide computer- and network-based methods and systems for two-way initiated, bi-directional communication with wireless devices using connection-based or connection-less protocols, such as, for example, TCP/IP and UDP/IP. Example embodiments provide an Address Management Proxy System (“AMPS”), which enables devices and systems connected to a public internetnetwork, such as the Internet, to initiate communication with and send data to wireless devices connected to a private wireless network, without exposing the non-routable private addresses of these wireless devices. The</p>

Exhibit C-29

Claims	VMware Systems
	<p>AMPS allocates a public (routable) network address for temporary use by a requesting device on an external public network to communicate with a wireless device on a private wireless network. In one embodiment, a pool of public addresses, for example, public IP addresses, is maintained by the AMPS and allocated dynamically to wireless network devices as required. The mapping of a temporary public address to the private address of a wireless device is maintained and updated transparently by the AMPS using routing tables and other mapping data structures.” Mehta at [0011].</p> <p>“To insure a greater degree of security, according to one embodiment, the AMPS maintains a Time to Live (TTL) parameter with each address mapping. In this way, once the TTL value indicates that the mapping has expired, the AMPS can destroy the mapping, and also any connection. Further, in some embodiments, the AMPS also puts a timestamp in its own mapping tables. After some timeout period based upon the timestamp, the AMPS can destroy the mapping, thereby forcing a new mapping to be initiated on a periodic basis.” Mehta at [0015].</p> <p>“Embodiments of the present invention provide computer- and network-based methods and systems for two-way initiated, bi-directional communication with wireless devices using connection-based or connection-less protocols, such as, for example, TCP/IP and UDP/IP. Example embodiments provide an Address Management Proxy System (“AMPS”), which enables devices and systems connected to a public internet, such as the Internet, to initiate communication with and to send data to wireless devices connected to a private wireless network, without exposing the non-routable private addresses of these wireless devices.” Mehta at [0024].</p> <p>“The second approach used to implement the AMPS supports full bi-directional communication through point-to-point connections established, for example, using TCP/IP protocol. (Note that these same techniques also support connection-less UDP bi-directional communication). The second approach can be implemented by providing a modified implementation of a standard UDP or TCP/IP function,</p>

Exhibit C-29

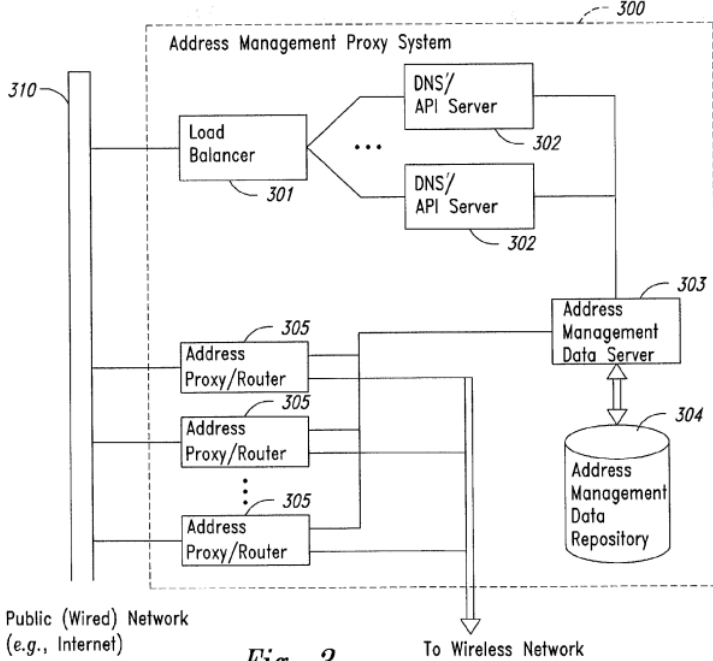
Claims	VMware Systems
	<p>“GetHostByName.” The GetHostByName API allows a string designation to identify the designated device and returns an IP address data structure. Alternatively, to increase the level of security provided, the AMPS can implement a specialized API to return the dynamically allocated public address that (now) corresponds to the requested wireless device. A disadvantage of the specialized API approach is that the device on the public network (or other device that wishes to obtain a connection to the wireless device) needs to include specialized code in the application on the requesting device.” Mehta at [0041].</p>  <p style="text-align: center;"><i>Fig. 3</i></p>

Exhibit C-29

Claims	VMware Systems
	<p>Mehta at FIG. 3.</p> <p>“Methods and systems consistent with the present invention establish a virtual network on top of current IP network naming schemes. The virtual network uses a separate layer to create a modification to the IP packet format that is used to separate network behavior from addressing. As a result of the modification to the packet format, any type of delivery method may be assigned to any address or group of addresses. The virtual network also maintains secure communications between nodes, while providing the flexibility of assigning delivery methods independent of the delivery addresses.” Caronni I at Abstract.</p> <p>“The present invention relates generally to data processing systems and, more particularly, to a private network using a public-network infrastructure.” Caronni I at 1:57–59.</p> <p>“Methods and systems consistent with the present invention overcome the shortcomings of existing networks by establishing a “Supernet,” which is a private network that uses components from a public-network infrastructure. A Supernet allows an organization to utilize a public-network infrastructure for its enterprise network so that the organization no longer has to maintain a private network infrastructure; instead, the organization may have the infrastructure maintained for them by one or more service providers or other organizations that specialize in such connectivity matters. As such, the burden of maintaining an enterprise network is greatly reduced. Moreover, a Supernet is not geographically restrictive, so a user may plug their device into the Internet from virtually any portal in the world and still be able to use the resources of their private network in a secure and robust manner.” Caronni I at 4:36-52.</p> <p>“Supernets also provide heterogeneous addressing functionality. The Supernet uses a separate layer that isolates address names of nodes from addressing schemes and delivery schemes. The Supernet contains a modification to the IP packet format that can be used to separate network behavior from addressing. As a result of the</p>

Exhibit C-29

Claims	VMware Systems
	<p>modification, any delivery scheme may be assigned to any address, or group of addresses.” Caronni I at 4:53-59.</p> <p><i>See also</i> Caronni I at Claim 1, FIGS. 3, 5.</p> <p>To the extent that VMware Systems does not disclose any elements of this claim limitation, it would have been obvious to a person having ordinary skill in the art to combine, with the teachings of VMware Systems , those elements, as known to them or as disclosed by other prior art identified in the cover pleading to Defendant’s Invalidation Contentions.</p>
<p>[30.1] a network interface configured for data communication via a virtual network that is defined by a domain name having an associated public network address;</p>	<p>VMware Systems discloses or renders obvious a network interface configured for data communication via a virtual network that is defined by a domain name having an associated public network address. For example:</p> <p>“Networking: Up to four virtual Ethernet NICs. Each virtual NIC may be high-performance VMware virtual NIC or AMD® PCnet™-PCI II compatible virtual NIC.” (VMware ESX Server Specifications)</p> <p>“Ethernet Card: Up to four virtual Ethernet cards. AMD® PCnet™ -PCI II compatible. PXE ROM version 2.0.” (VMware GSX Server 3.1 Administration Guide)</p> <p>“Virtual Networking and File Sharing: Nine virtual Ethernet switches (three reserved for bridged, host-only and NAT networking).” (VMware GSX Server Specifications)</p> <p>To the extent this reference does not expressly or inherently disclose this limitation, it would have been obvious in view of the background knowledge of a person of ordinary skill in the art. This reference further renders the limitation obvious in light of other references, such as Mehta and/or Caronni I. Mehta and Caronni I concern similar subject matter as this reference, namely, virtual networks. As set forth in the accompanying contentions, use of virtual networks was well-known and common.</p>

Exhibit C-29

Claims	VMware Systems
	<p>Modifying this reference with Mehta’s and/or Caronni I’s teachings would have only required simple substitution, would not have required extensive experimentation, and would have yielded predictable results. A person of ordinary skill in the art would have further understood how to modify the disclosures of this reference with that of Mehta or Caronni I for example, to incorporate this limitation. <i>See, e.g.,</i></p> <p>“The present invention relates to methods and systems for initiating communication with wireless devices and, in particular, to methods and systems for initiating communication with a device on a private network from a device on a public network to achieve virtual end-to-end connectivity.” Mehta at [0002].</p> <p>“Embodiments of the present invention provide computer- and network-based methods and systems for two-way initiated, bi-directional communication with wireless devices using connection-based or connection-less protocols, such as, for example, TCP/IP and UDP/IP. Example embodiments provide an Address Management Proxy System (“AMPS”), which enables devices and systems connected to a public internetnetwork, such as the Internet, to initiate communication with and send data to wireless devices connected to a private wireless network, without exposing the non-routable private addresses of these wireless devices. The AMPS allocates a public (routable) network address for temporary use by a requesting device on an external public network to communicate with a wireless device on a private wireless network. In one embodiment, a pool of public addresses, for example, public IP addresses, is maintained by the AMPS and allocated dynamically to wireless network devices as required. The mapping of a temporary public address to the private address of a wireless device is maintained and updated transparently by the AMPS using routing tables and other mapping data structures.” Mehta at [0011].</p> <p>“Although the techniques of the AMPS are generally applicable to any a wired device communicating with a wireless device, the phrase “public network” (or “wired network”) is used generally to imply any type of internetworked</p>

Exhibit C-29

Claims	VMware Systems
	<p>environment including a public network or a backbone that is somewhere down the line connected to one or more private or public networks. In addition, although the examples described herein often refer to the Internet, one skilled in the art will recognize that the concepts and inventions described are applicable to other forms and embodiments of internetworking, including, for example ATM type networks. Thus, techniques of the present invention can also be used by one device on a first wireless network to communicate with another wireless device on a second network—each device ends up communicating with the Address Proxy/Router of the other network. This scenario is feasible because each wireless network (or its carrier infrastructure) is connected to a proxy/router that is also connected (via a public address) to a public network. In addition, although a public network is sometimes also referred to herein as a wired network, one skilled in the art will recognize that any network that exposes routable (public) addresses may be implied. Thus, a wireless network with unique public (and routable) address can also employ techniques of the present invention to perform bi-directional communication. Also, one skilled in the art will recognize that terms such as wireless device, phone, handheld, etc., are used interchangeably to indicate any type of wireless device that is capable of operating with the AMPS. In addition, terms may have alternate spellings which may or may not be explicitly mentioned, and one skilled in the art will recognize that all such variations of terms are intended to be included.” Mehta at [0033].</p> <p>“Public address to proxy/router machine table 620 comprises a public network address field 631 and an indication of a functioning proxy/router machine 621. By maintaining such a mapping, the AMPS is able to substitute proxy/router machines for other proxy/router machines to provide a higher degree of robustness. Each proxy/router machine has a preconfigured set of public network addresses, such as are typically configured by network cards inserted into the proxy/router machine. These address are allocated in a standard fashion through prior purchase or obtaining from an address authorizing authority, currently the Internet Corporation for Assigned Names and Numbers (ICANN). When a machine is inserted for use</p>

Exhibit C-29

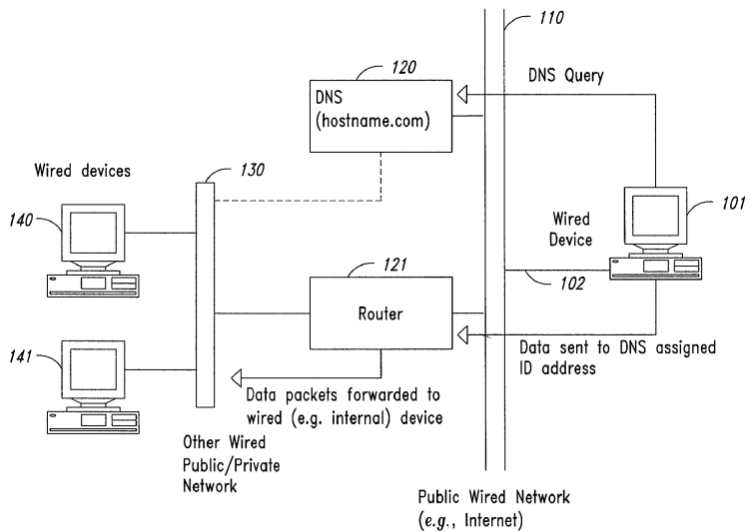
Claims	VMware Systems
	<p>into the AMPS, indications of these addresses are stored in field 631.” Mehta at [0055].</p>  <p>The diagram illustrates a network configuration. On the left, a vertical line represents a network boundary. To its left is an 'Other Wired Public/Private Network' containing two computer icons labeled 140 and 141. A vertical line labeled 130 represents a switch or hub connecting these devices. To the right of the boundary is a 'Public Wired Network (e.g., Internet)'. A box labeled 120, 'DNS (hostname.com)', is connected to the boundary. A box labeled 121, 'Router', is also connected to the boundary. A computer icon labeled 101, 'Wired Device', is connected to the boundary via a line labeled 102. An arrow labeled 'DNS Query' points from the wired device to the DNS server. An arrow labeled 'Data sent to DNS assigned ID address' points from the wired device to the router. An arrow labeled 'Data packets forwarded to wired (e.g. internal) device' points from the router to the internal network switch.</p> <p style="text-align: center;"><i>Fig. 1</i></p> <p>Mehta at FIG. 1.</p>

Exhibit C-29

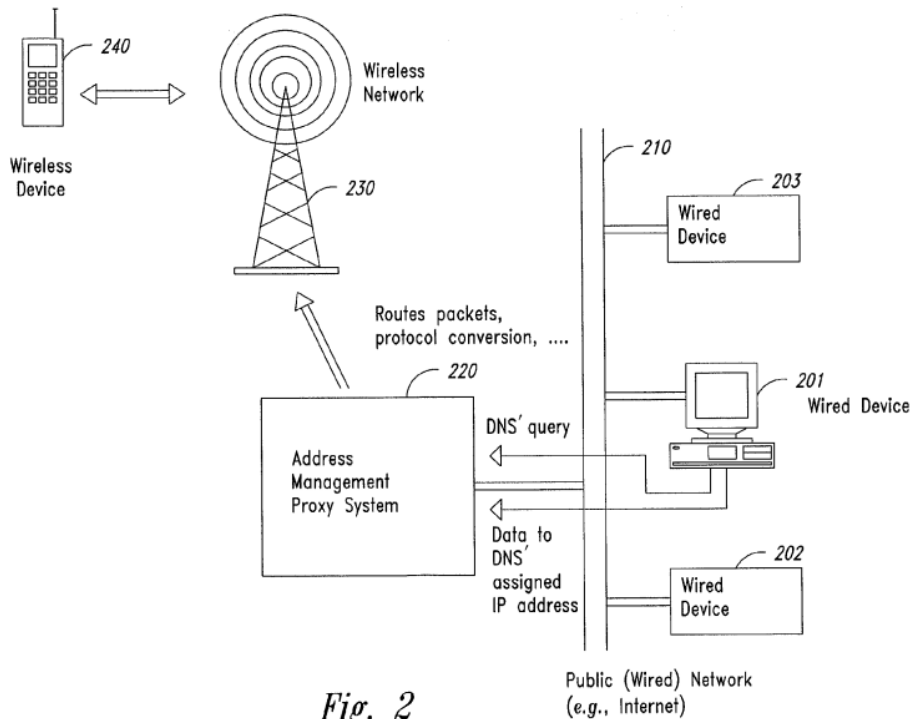
Claims	VMware Systems
	 <p style="text-align: center;"><i>Fig. 2</i></p> <p>Mehta at FIG. 2.</p> <p>“Methods and systems consistent with the present invention establish a virtual network on top of current IP network naming schemes. The virtual network uses a separate layer to create a modification to the IP packet format that is used to separate network behavior from addressing. As a result of the modification to the packet format, any type of delivery method may be assigned to any address or group of addresses. The virtual network also maintains secure communications</p>

Exhibit C-29

Claims	VMware Systems
	<p>between nodes, while providing the flexibility of assigning delivery methods independent of the delivery addresses.” Caronni I at Abstract.</p> <p>“The present invention relates generally to data processing systems and, more particularly, to a private network using a public-network infrastructure.” Caronni I at 1:57–59.</p> <p>“Methods and systems consistent with the present invention overcome the shortcomings of existing networks by establishing a “Supernet,” which is a private network that uses components from a public-network infrastructure. A Supernet allows an organization to utilize a public-network infrastructure for its enterprise network so that the organization no longer has to maintain a private network infrastructure; instead, the organization may have the infrastructure maintained for them by one or more service providers or other organizations that specialize in such connectivity matters. As such, the burden of maintaining an enterprise network is greatly reduced. Moreover, a Supernet is not geographically restrictive, so a user may plug their device into the Internet from virtually any portal in the world and still be able to use the resources of their private network in a secure and robust manner.” Caronni I at 4:36-52.</p> <p>“Supernets also provide heterogeneous addressing functionality. The Supernet uses a separate layer that isolates address names of nodes from addressing schemes and delivery schemes. The Supernet contains a modification to the IP packet format that can be used to separate network behavior from addressing. As a result of the modification, any delivery scheme may be assigned to any address, or group of addresses.” Caronni I at 4:53-59.</p> <p>“SNlogin 522 is a script used for logging into a Supernet. Successfully executing this script results in a Unix shell from which programs (e.g., node A 522) can be started to run within the Supernet context, such that address translation and security encapsulation is performed transparently for them and all they can typically access is other nodes on the Supernet. Alternatively, a parameter may be passed into</p>

Exhibit C-29

Claims	VMware Systems
	<p>SNlogin 522 that indicates a particular process to be automatically run in a Supernet context. Once a program is running in a Supernet context, all programs spawned by that program also run in the Supernet context, unless explicitly stated otherwise. SNlogout 524 is a script used for logging out of a Supernet. Although both SNlogin 522 and SNlogout 524 are described as being scripts, one skilled in the art will appreciate that their processing may be performed by another form of software. VARPD 526 performs address translation between node IDs and real addresses. KMC 528 is the key management component for each node that receives updates whenever the key for a channel (“the channel key”) changes. There is one KMC per node per channel. KMD 530 receives requests from SNSL 542 of the TCP/IP protocol stack 534 when a packet is received and accesses the appropriate KMC for the destination node to retrieve the appropriate key to decrypt the packet. Node A 532 is a Supernet node running in a Supernet context.” Caronni I at 8:31-55.</p> <p>“TCP/IP protocol stack 534 contains a standard TCP/UDP layer 538, two standard IP layers (an inner IP layer 540 and an outer IP layer 544), and a Supernet security layer (SNSL) 542, acting as the conduit for all Supernet communications. To conserve memory, both inner IP layer 540 and outer IP layer 544 may share the same instance of the code of an IP layer. SNSL 542 performs security functionality as well as address translation. It also caches the most recently used channel keys for ten seconds. Thus, when a channel key is needed, SNSL 542 checks its cache first, and if it is not found, it requests KMD 530 to contact the appropriate KMC to retrieve the appropriate channel key.” Caronni I at 8:54-67.</p> <p>“FIG. 5 depicts administrative machine 306 and device 302 in greater detail, although the other devices 304 and 308–312 may contain similar components. Device 302 and administrative machine 306 communicate via Internet 314. Each device contains similar components, including a memory 502, 504; secondary storage 506, 508; a central processing unit (CPU) 510, 512; an input device 514, 516; and a video display 518, 520. One skilled in the art will appreciate that these devices may contain additional or different components. Memory 504 of</p>

Exhibit C-29

Claims	VMware Systems
	<p>administrative machine 306 includes the SASD process 540, VARPD 548, and KMS 550 all running in user mode. That is, CPU 512 is capable of running in at least two modes: user mode and kernel mode. When CPU 512 executes programs running in user mode, it prevents them from directly manipulating the hardware components, such as video display 518. On the other hand, when CPU 512 executes programs running in kernel mode, it allows them to manipulate the hardware components. Memory 504 also contains a VARPDB 551 and a TCP/IP protocol stack 552 that are executed by CPU 512 running in kernel mode. TCP/IP protocol stack 552 contains a TCP/UDP layer 554 and an IP layer 556, both of which are standard layers well known to those of ordinary skill in the art. Secondary storage 508 contains a configuration file 558 that stores various configuration-related information (described below) for use by SASD 540.” Caronni I at 6:44-7:2.</p> <p><i>See also Caronni I at Claim 1, FIGS. 3, 5.</i></p> <p>To the extent that VMware Systems does not disclose any elements of this claim limitation, it would have been obvious to a person having ordinary skill in the art to combine, with the teachings of VMware Systems , those elements, as known to them or as disclosed by other prior art identified in the cover pleading to Defendant’s Invalidation Contentions.</p>
<p>[30.2] a memory and a processor to implement a register module configured to register devices in a virtual network,</p>	<p>VMware Systems discloses or renders obvious a memory and a processor to implement a register module configured to register devices in a virtual network. For example:</p> <p>“VMware GSX Server guarantees server resources for CPU, memory, network bandwidth, and disk I/O at optimum performance levels.” (VMware GSX Server 3.1 Virtual Machine Guide)</p> <p>“The VMware Virtual Infrastructure SDK provides comprehensive interfaces to: Create, delete, copy and clone virtual machines.” (VMware Delivers Open Interface to Virtual Infrastructure)</p>

Exhibit C-29

Claims	VMware Systems
	<p data-bbox="814 261 1812 358">“The RegisterVm method registers a virtual machine on a server where vmName is a string specifying the virtual machine’s configuration file name.” (VMware Scripting API User’s Manual)</p> <p data-bbox="751 394 1797 833">To the extent this reference does not expressly or inherently disclose this limitation, it would have been obvious in view of the background knowledge of a person of ordinary skill in the art. This reference further renders the limitation obvious in light of other references, such as Mehta, Caronni I, Caronni II, and/or Hipp. Mehta, Caronni I, Caronni II, and Hipp concern similar subject matter as this reference, namely, virtual networks. As set forth in the accompanying contentions, registration of devices in a virtual network was a well-known and commonly-used technique. Modifying this reference with Mehta’s, Caronni I’s, Caronni II’s, and/or Hipp’s teachings would have only required simple substitution, would not have required extensive experimentation, and would have yielded predictable results. A person of ordinary skill in the art would have further understood how to modify the disclosures of this reference with those of Mehta, Caronni I, Caronni II, or Hipp for example, to incorporate this limitation. <i>See, e.g.,</i></p> <p data-bbox="814 902 1812 1341">“Methods and systems for providing two-way initiated, bi-directional communication with wireless devices using connection-based or connection-less protocols, such as, for example, TCP/IP and UDP/IP, are provided. Example embodiments provide an Address Management Proxy System (“AMPS”), which enables devices and systems connected to a public internet, such as the Internet, to initiate communication with and to send data to wireless devices connected to a private wireless network, without exposing the non-routable private addresses of these wireless devices. The AMPS allocates a public (routable) network address for temporarily use by a requesting device on a public network to communicate with a wireless device on a wireless network. In one embodiment, a pool of public addresses, for example, public IP addresses, is maintained by the AMPS and allocated dynamically to wireless network devices as required. In one embodiment, the AMPS comprises one or more modified DNS/API servers, one or more Address</p>

Exhibit C-29

Claims	VMware Systems
	<p>Proxy/Routers, an Address Management Data Server, one or more data repositories, and optionally a load balancer. The AMPS DNS/API server receives a request from a device on a public network for a particular wireless device, and returns an appropriate temporary public address, which is internally mapped to the private address of the wireless device. The public address is then usable by the device on the public network to send data to the wireless device.” Mehta at Abstract.</p> <p>“To insure a greater degree of security, according to one embodiment, the AMPS maintains a Time to Live (TTL) parameter with each address mapping. In this way, once the TTL value indicates that the mapping has expired, the AMPS can destroy the mapping, and also any connection. Further, in some embodiments, the AMPS also puts a timestamp in its own mapping tables. After some timeout period based upon the timestamp, the AMPS can destroy the mapping, thereby forcing a new mapping to be initiated on a periodic basis.” Mehta at [0015].</p> <p>“In existing systems, data communication (communication on a data channel) between a wireless device that is connected to a private wireless network and a wired device connected to a public wired network (e.g., the Internet) can only be initiated by the wireless device. Some carriers have assigned fixed public IP addresses to wireless devices; however, the wireless devices need to then have client programs (e.g., a UDP stack) capability of receiving and handling the incoming communication packets. Moreover, these wireless devices are then part of a public wireless network and not a private wireless network. Since public IP addresses are becoming a scarcer commodity and currently expensive to service a network of multi-million devices, carriers cannot in a practical sense count on having a fixed public IP address for each device on its network. (Although movement to IPv6 will allow more addresses, the current IPv4 definitions are limited and the potential number of wireless users subscribing to larger carriers is very high. In some regions of the world, the current public address scheme is even more limited.) Further, such addressing capability would expose each device to further security risks, because each such device is part of a publicly accessible</p>

Exhibit C-29

Claims	VMware Systems
	<p>network and it becomes more difficult to implement and enforce security measures. Thus, assigning private IP addresses to devices is preferred in existing wireless networks over assigning fixed public IP addresses to devices. When wireless networks are private, the locations (addresses) of the wireless devices are intentionally hidden from public view by a carrier system's (or, as referred to in some countries, operator's) infrastructure.” Mehta at [0025].</p> <p>“Example embodiments described herein provide applications, tools, data structures and other support to implement private to public address mappings over one or more wired and wireless networks to be used for bi-directional communication. One skilled in the art will recognize that other embodiments of the methods and systems of the present invention may be used for many other purposes, including to push information and/or data or code from a public network such as the Internet to a wireless device. In addition, although this description primarily refers to “data” as being sent via the networks, one skilled in the art will recognize that all types of data can be communicated using the techniques described herein including, but not limited to, text, graphics, audio, and video.” Mehta at [0034].</p> <p>“FIG. 4 is an example block diagram of a general purpose computer system for practicing embodiments of the Address Management Proxy System. The general purpose computer system 400 may comprise one or more server and/or client computing systems and may span distributed locations. In addition, each block may represent one or more such blocks as appropriate to a specific embodiment or may be combined with other blocks. The various blocks of the Address Management Proxy System 410 may physically reside on one or more machines, which use standard interprocess communication mechanisms to communicate with each other. In the embodiment shown, computer system 400 comprises a computer memory (“memory”) 01, a display 402, a Central Processing Unit (“CPU”) 403, and Input/Output devices 404. The Address Management Proxy System 410 is shown residing in memory 401. The components of the Address Management Proxy System 410 preferably execute on CPU 403 and manage the address mapping of</p>

Exhibit C-29

Claims	VMware Systems
	<p>wireless devices on a wireless network, as described in previous figures, to allow other wired systems to communicate with the wireless devices. Other downloaded code 405 and potentially other data repositories also reside in the memory 410, and preferably execute on one or more CPU's 403. In a typical embodiment, the AMPS 410 includes one or more DNS/API servers 411, one or more Address Proxy/Routers 412, an Address Management Data Server 413, and Address Management Data Repositories 414. As described earlier, the AMPS may include other data repositories and components, such as a load balancer, depending upon the particular implementation.” Mehta at [0036].</p> <p>“In one embodiment, a timestamp of the mapping between a public network address and a private address is also noted in table 630. After a defined time out, based upon this timestamp, the Address Management Data Server sends a request to the Address Proxy/Router associated with the mapping to unmap the public-to-private address mapping and updates the mapping table 630, thereby returning the associated public address to the pool of unused public network addresses.” Mehta at [0056].</p>

Exhibit C-29

Claims	VMware Systems
	<div data-bbox="919 289 1650 906" data-label="Diagram"> <p>The diagram, labeled Fig. 4, illustrates a Computer System (400). It features a Memory section (401) which contains several components: a DNS/API Server (411), an Address Proxy/Router (412), an Address Management Data Server (413), and an Address Management Data Repository (414). The Address Management Data Server (413) and the Address Management Data Repository (414) are connected to a component labeled AMPS. Additionally, there is a block for Other Code (405) within the memory section. Below the memory section, the system includes a Display (402), a CPU (403), and Input/Output Devices (404).</p> </div> <p data-bbox="1171 959 1283 1003"><i>Fig. 4</i></p> <p data-bbox="814 1062 1016 1089">Mehta at FIG. 4.</p> <p data-bbox="804 1159 1808 1326">“To configure a Supernet, a system administrator creates a configuration file 558 that is used by SASD 540 when starting or reconfiguring a Supernet. This file may specify: (1) the Supernet name, (2) all of the channels in the Supernet, (3) the nodes that communicate over each channel, (4) the address of the KMS for each channel, (5) the address of the VARPD that acts as the server for the Supernet, (6) the user</p>

Exhibit C-29

Claims	VMware Systems
	<p>IDs of the users who are authorized to create Supernet nodes, (7) the authentication mechanism to use for each user of each channel, and (8) the encryption algorithm to use for each channel. Although the configuration information is described as being stored in a configuration file, one skilled in the art will appreciate that this information may be retrieved from other sources, such as databases or interactive configurations.” Caronni I at 8:1-15.</p> <p>“SASD 540 represents a Supernet: there is one instance of an SASD per Supernet, and it both authenticates nodes and authorizes nodes to join the Supernet. VARPD 548 has an associated component, VARPDB 551, into which it stores mappings of the internal Supernet addresses, known as a node IDs, to the network addresses recognized by the public-network infrastructure, known as the real addresses. The “node ID” may include the following: a Supernet ID (e.g., 0x123), reflecting a unique identifier of the Supernet, and a virtual address, comprising an IP address (e.g., 10.0.0.1). Although the virtual address is described in an IP address scheme, one skilled in the art will appreciate that the virtual address may be any other type addressing scheme, such as an e-mail address, IPX, or IPv6. Since the node ID includes a Supernet ID, a node will have more than one node ID when it communicates over more than one channel. The “real address” is an IP address (e.g., 10.0.0.2) that is globally unique and meaningful to the public-network infrastructure. In a Supernet, one VARPD runs on each machine, and it may play two roles. First, a VARPD may act as a server by storing all address mappings for a particular Supernet into its associated VARPDB. Second, regardless of its role as a server or not, each VARPD assists in address translation for the nodes on its machine. In this role, the VARPD stores into its associated VARPDB the address mappings for its nodes, and if it needs a mapping that it does not have, it will contact the VARPD that acts as the server for the given Supernet to obtain it. The VARPDB may also decide which virtual address to use in the translation. That is, the VARPDB may associate a virtual address with multiple real addresses or vice versa.” Caronni I at 7:3–33.</p>

Exhibit C-29

Claims	VMware Systems
	<p>“FIGS. 7A and 7B depict a flow chart of the steps performed when a node joins a Supernet. The first step performed is that the user invokes the SNlogin script and enters the Supernet name, their user ID, their password, and a requested virtual address (step 702). Of course, this information depends on the particular authentication mechanism used. Upon receiving this information, the SNlogin script performs a handshaking with SASD to authenticate this information. In this step, the user may request a particular virtual address to be used, or alternatively, the SASD may select one for them. Next, if any of the information in step 702 is not validated by SASD (step 704), processing ends. Otherwise, upon successful authentication, SASD creates an address mapping between a node ID and the real address (step 706). In this step, SASD concatenates the Supernet ID with the virtual address to create the node ID, obtains the real address of the SNlogin script by querying network services in a well-known manner, and then registers this information with the VARPd that acts as the server for this Supernet. This VARPd is identified in the configuration file.” Caronni I at 9:67–10:18.</p> <p>“After configuring SNSL, SNlogin invokes an operating system call, SETVIN, to cause the SNlogin script to run in a Supernet context (step 720). In Unix, each process has a data structure known as the “proc structure” that contains the process ID as well as a pointer to a virtual memory description of this process. In accordance with methods and systems consistent with the present invention, the Supernet IDs indicating the channels over which the process communicates as well as its virtual address for this process are added to this structure. By associating this information with the process, the SNSL layer can enforce that this process runs in a Supernet context. Although methods and systems consistent with the present invention are described as operating in a Unix environment, one skilled in the art will appreciate that such methods and systems can operate in other environments. After the SNlogin script runs in the Supernet context, the SNlogin script spawns a Unix program, such as a Unix shell or a service daemon (step 722). In this step, the SNlogin script spawns a Unix shell from which programs can be run by the user. All of these</p>

Exhibit C-29

Claims	VMware Systems
	<p>programs will thus run in the Supernet context until the user runs the SNlogout script.” Caronni I at 10:63–11:17.</p> <p>“The packet and Supernet ID are then transmitted to the SNSL layer using the modified socket structure (step 806). The SNSL layer then accesses the VARPDB to obtain the address mapping between virtual source node address 642 and the source real address 614 as well as the virtual destination node address 644 and the destination real address 616 (step 808). If they are not contained in the VARPDB because this is the first time a packet has been sent from this node or sent to this destination, the VARPDB accesses the local VARPD to obtain the mapping. When contacted, the VARPD on the local machine contacts the VARPD that acts as the server for the Supernet to obtain the appropriate address mapping. Since the VARPDB maintains all real IP addresses, a remote node may securely communicate with another remote node without reverfication.” Caronni I at 11:47–61.</p> <p><i>See also</i> Caronni I at FIGS. 7A and 7B.</p> <p>“In an embodiment of the present invention a virtual network is supported by a physical network and includes a virtual address resolution facility. The virtual address resolution facility is used to resolve a virtual IP address into a real IP address. A registration request is received at the virtual address resolution facility which includes a real IP address, a port address, a transport protocol designation and an Application layer protocol designation associated with a virtual address destination. The associations are stored using the virtual address resolution facility. A resolution request is received referencing a virtual address destination and the resolution request is resolved using the virtual address resolution facility and the stored associations.” Caronni II at 3:11–23.</p> <p>“FIG. 2 depicts a block diagram of the VARP lookup table 26 used by the virtual address resolution facility 24. The VARP lookup table 26 may be stored at any</p>

Exhibit C-29

Claims	VMware Systems
	<p>location accessible over the network. The VARP lookup table 26 includes registered virtual destination addresses located in a column 78 and corresponding associations located in a column 79. Entries 80, 82, 84, 86 and 88 include the registered virtual destination address and information associated with the registered virtual destination address. Each entry in the VARP lookup table 26 will list the registering virtual IP address 90 and associated information including a real IP address 91, a transport protocol designation 92, a port number identification 93 and an Application layer protocol designation 94. The associations are provided at the time the virtual destination address is registered in the virtual network. For example, the entry 80 for the virtual IP address 10.0.0.12 (90) includes the association of a real IP address 152.70.0.1 (91), an associated Transport layer protocol designation, UDP 92, an associated port number, 6789 (93), and an Application layer protocol designation of ‘none’ 94.” Caronni II at 5:10–29.</p> <p>“FIG. 4 depicts the sequence of steps followed by the illustrative embodiment of the present invention to process the received encapsulated message. The sequence begins when a process associated with a virtual IP address registers with the virtual address resolution facility 24 (step 140). The registration is stored in a VARP lookup table accessible over the network. Subsequently, a message is received at the electronic device which bears a MAC/network interface address of the network interface of the electronic device with the designated real IP address listed in the virtual address registration (Step 142). The MAC address header 117 is stripped off at the Link layer and the message and appended headers are passed up to the Network layer 114(Step 144). The Network layer 114 strips off the IP header 115 and identifies the Transport Protocol header 113 underneath. The Network layer passes the message and remaining appended header to the Transport layer 112 (Step 146). The Transport layer 112 strips off the Transport header and then passes the message to the Application layer via the port address of the interface identified in the VARP registration. (Step 148). The receiving process determines whether it is the virtual destination address the message is addressed to (Step 149) and acts accordingly. If the receiving process that is examining the message is the destination</p>

Exhibit C-29

Claims	VMware Systems
	<p>address, the process examines the data (Step 150). Alternatively, if the message is not intended for the process executing on the edge device, the process re-routes the message and sends it back down the protocol model stack, on to the local network operating behind the edge device, and on to its intended destination by performing traditional VARP and ARP lookups. Alternatively forwarding the message may be accomplished by reinserting the packet into the IP layer of the networking stack.” Caronni II at 7:33–67.</p> <p><i>See also</i> Caronni II at Claim 1, FIGS. 2-4.</p> <p>“FIG. 3 is a data flow diagram illustrating the registration of virtual network environment parameters. The VNE framework 200 is a software module that processes transactions between the applications and the operating system. The VNE parameters are registered with the VNE framework 200 at the time the application is started. The VNE parameters include the application IP address, the virtual network subnet, and the global virtual address subnet. At step 250, the registration harness 220 supplies the application IP address, virtual subnet, and the global virtual address subnet for a process_x to the VNE framework 200. The VNE framework 200 then records the IP address, virtual subnet, and the global virtual address subnet for the process_x at step 252. The process_x can then spawn additional processes (or create additional objects) at step 254. the new process_y inherits the IP address, virtual subnet, and global virtual address subnet from process_x. At step 256, the registration harness 220 launches the application related to process_y.” Hipp at 3:52-4:2.</p> <p>“Virtualization of network identity is achieved by assigning a unique virtual IP address and virtual hostname to a group of processes that make up the application instance which the instance keeps throughout its execution. This virtual network identity stays with the application instance regardless of which node the application is running on. The framework, in essence, provides a mechanism to create this virtual network identity (VNI) around the application using the virtual network</p>

Exhibit C-29

Claims	VMware Systems
	<p>parameters assigned to it. In one embodiment, the virtual network parameters include an IP address and hostname. The framework 200 ensures that the application's instance uses the virtual network parameters, transparently, so that it can be moved across machines, without modifications to the application.” Hipp at 6:1-15.</p> <p><i>See also</i> Hipp at 6:16-60, FIG. 3.</p> <p>To the extent that VMware Systems does not disclose any elements of this claim limitation, it would have been obvious to a person having ordinary skill in the art to combine, with the teachings of VMware Systems , those elements, as known to them or as disclosed by other prior art identified in the cover pleading to Defendant’s Invalidation Contentions.</p>
<p>[30.3] the register module further configured to: receive a registration request from an agent associated with a device;</p>	<p>VMware Systems discloses or renders obvious the register module further configured to: receive a registration request from an agent associated with a device.</p> <p>To the extent this reference does not expressly or inherently disclose this limitation, it would have been obvious in view of the background knowledge of a person of ordinary skill in the art. This reference further renders the limitation obvious in light of other references, such as Mehta, Caronni I, Caronni II, and/or Hipp. Mehta, Caronni I, Caronni II, and Hipp concern similar subject matter as this reference, namely, virtual networks. As set forth in the accompanying contentions, registration of devices in a virtual network was a well-known and commonly-used technique. Modifying this reference with Mehta’s, Caronni I’s, Caronni II’s, and/or Hipp’s teachings would have only required simple substitution, would not have required extensive experimentation, and would have yielded predictable results. A person of ordinary skill in the art would have further understood how to modify the disclosures of this reference with those of Mehta, Caronni I, Caronni II, or Hipp for example, to incorporate this limitation. <i>See, e.g.,</i></p>

Exhibit C-29

Claims	VMware Systems
	<p>“Methods and systems for providing two-way initiated, bi-directional communication with wireless devices using connection-based or connection-less protocols, such as, for example, TCP/IP and UDP/IP, are provided. Example embodiments provide an Address Management Proxy System (“AMPS”), which enables devices and systems connected to a public internet, such as the Internet, to initiate communication with and to send data to wireless devices connected to a private wireless network, without exposing the non-routable private addresses of these wireless devices. The AMPS allocates a public (routable) network address for temporarily use by a requesting device on a public network to communicate with a wireless device on a wireless network. In one embodiment, a pool of public addresses, for example, public IP addresses, is maintained by the AMPS and allocated dynamically to wireless network devices as required. In one embodiment, the AMPS comprises one or more modified DNS/API servers, one or more Address Proxy/Routers, an Address Management Data Server, one or more data repositories, and optionally a load balancer. The AMPS DNS/API server receives a request from a device on a public network for a particular wireless device, and returns an appropriate temporary public address, which is internally mapped to the private address of the wireless device. The public address is then usable by the device on the public network to send data to the wireless device.” Mehta at Abstract.</p> <p>“To insure a greater degree of security, according to one embodiment, the AMPS maintains a Time to Live (TTL) parameter with each address mapping. In this way, once the TTL value indicates that the mapping has expired, the AMPS can destroy the mapping, and also any connection. Further, in some embodiments, the AMPS also puts a timestamp in its own mapping tables. After some timeout period based upon the timestamp, the AMPS can destroy the mapping, thereby forcing a new mapping to be initiated on a periodic basis.” Mehta at [0015].</p> <p>“In existing systems, data communication (communication on a data channel) between a wireless device that is connected to a private wireless network and a wired device connected to a public wired network (e.g., the Internet) can only be</p>

Exhibit C-29

Claims	VMware Systems
	<p>initiated by the wireless device. Some carriers have assigned fixed public IP addresses to wireless devices; however, the wireless devices need to then have client programs (e.g., a UDP stack) capability of receiving and handling the incoming communication packets. Moreover, these wireless devices are then part of a public wireless network and not a private wireless network. Since public IP addresses are becoming a scarcer commodity and currently expensive to service a network of multi-million devices, carriers cannot in a practical sense count on having a fixed public IP address for each device on its network. (Although movement to IPv6 will allow more addresses, the current IPv4 definitions are limited and the potential number of wireless users subscribing to larger carriers is very high. In some regions of the world, the current public address scheme is even more limited.) Further, such addressing capability would expose each device to further security risks, because each such device is part of a publicly accessible network and it becomes more difficult to implement and enforce security measures. Thus, assigning private IP addresses to devices is preferred in existing wireless networks over assigning fixed public IP addresses to devices. When wireless networks are private, the locations (addresses) of the wireless devices are intentionally hidden from public view by a carrier system's (or, as referred to in some countries, operator's) infrastructure.” Mehta at [0025].</p> <p>“Example embodiments described herein provide applications, tools, data structures and other support to implement private to public address mappings over one or more wired and wireless networks to be used for bi-directional communication. One skilled in the art will recognize that other embodiments of the methods and systems of the present invention may be used for many other purposes, including to push information and/or data or code from a public network such as the Internet to a wireless device. In addition, although this description primarily refers to “data” as being sent via the networks, one skilled in the art will recognize that all types of data can be communicated using the techniques described herein including, but not limited to, text, graphics, audio, and video.” Mehta at [0034].</p>

Exhibit C-29

Claims	VMware Systems
	<p>“In one embodiment, a timestamp of the mapping between a public network address and a private address is also noted in table 630. After a defined time out, based upon this timestamp, the Address Management Data Server sends a request to the Address Proxy/Router associated with the mapping to unmap the public-to-private address mapping and updates the mapping table 630, thereby returning the associated public address to the pool of unused public network addresses.” Mehta at [0056].</p> <p>“FIG. 3 depicts a data processing system 300 suitable for use with methods and systems consistent with the present invention. Data processing system 300 comprises a number of devices, such as computers 302–312, connected to a public network, such as the Internet 314. A Supernet's infrastructure uses components from the Internet because devices 302, 304, and 312 contain nodes that together form a Supernet and that communicate by using the infrastructure of the Internet. These nodes 316, 318, 320, and 322 are communicative entities (e.g., processes) running within a particular device and are able to communicate among themselves as well as access the resources of the Supernet in a secure manner. When communicating among themselves, the nodes 316, 318, 320, and 322 serve as end points for the communications, and no other processes or devices that are not part of the Supernet are able to communicate with the Supernet's nodes or utilize the Supernet's resources. The Supernet also includes an administrative node 306 to administer to the needs of the Supernet.” Caronni I at 4:66–5:13.</p> <p>“FIG. 5 depicts administrative machine 306 and device 302 in greater detail, although the other devices 304 and 308–312 may contain similar components. Device 302 and administrative machine 306 communicate via Internet 314. Each device contains similar components, including a memory 502, 504; secondary storage 506, 508; a central processing unit (CPU) 510, 512; an input device 514, 516; and a video display 518, 520. One skilled in the art will appreciate that these devices may contain additional or different components. Memory 504 of administrative machine 306 includes the SASD process 540, VARPD 548, and</p>

Exhibit C-29

Claims	VMware Systems
	<p>KMS 550 all running in user mode. That is, CPU 512 is capable of running in at least two modes: user mode and kernel mode. When CPU 512 executes programs running in user mode, it prevents them from directly manipulating the hardware components, such as video display 518. On the other hand, when CPU 512 executes programs running in kernel mode, it allows them to manipulate the hardware components. Memory 504 also contains a VARPDB 551 and a TCP/IP protocol stack 552 that are executed by CPU 512 running in kernel mode. TCP/IP protocol stack 552 contains a TCP/UDP layer 554 and an IP layer 556, both of which are standard layers well known to those of ordinary skill in the art. Secondary storage 508 contains a configuration file 558 that stores various configuration-related information (described below) for use by SASD 540.” Caronni I at 6:43–7:2.</p> <p>“Memory 502 of device 302 contains SNlogin script 522, SNlogout script 524, VARPD 526, KMC 528, KMD 530, and node A 532, all running in user mode. Memory 502 also includes TCP/IP protocol stack 534 and VARPDB 536 running in kernel mode.” Caronni I at 8:26-30.</p> <p>“FIGS. 7A and 7B depict a flow chart of the steps performed when a node joins a Supernet. The first step performed is that the user invokes the SNlogin script and enters the Supernet name, their user ID, their password, and a requested virtual address (step 702). Of course, this information depends on the particular authentication mechanism used. Upon receiving this information, the SNlogin script performs a handshaking with SASD to authenticate this information. In this step, the user may request a particular virtual address to be used, or alternatively, the SASD may select one for them. Next, if any of the information in step 702 is not validated by SASD (step 704), processing ends. Otherwise, upon successful authentication, SASD creates an address mapping between a node ID and the real address (step 706). In this step, SASD concatenates the Supernet ID with the virtual address to create the node ID, obtains the real address of the SNlogin script by querying network services in a well-known manner, and then registers this</p>

Exhibit C-29

Claims	VMware Systems
	<p>information with the VARP that acts as the server for this Supernet. This VARP is identified in the configuration file.” Caronni I at 9:66-10:18.</p> <p><i>See also</i> Caronni I at FIGS. 3, 5, 7A, 7B.</p> <p>“In an embodiment of the present invention a virtual network is supported by a physical network and includes a virtual address resolution facility. The virtual address resolution facility is used to resolve a virtual IP address into a real IP address. A registration request is received at the virtual address resolution facility which includes a real IP address, a port address, a transport protocol designation and an Application layer protocol designation associated with a virtual address destination. The associations are stored using the virtual address resolution facility. A resolution request is received referencing a virtual address destination and the resolution request is resolved using the virtual address resolution facility and the stored associations.” Caronni II at 3:11–23.</p> <p>“FIG. 2 depicts a block diagram of the VARP lookup table 26 used by the virtual address resolution facility 24. The VARP lookup table 26 may be stored at any location accessible over the network. The VARP lookup table 26 includes registered virtual destination addresses located in a column 78 and corresponding associations located in a column 79. Entries 80, 82, 84, 86 and 88 include the registered virtual destination address and information associated with the registered virtual destination address. Each entry in the VARP lookup table 26 will list the registering virtual IP address 90 and associated information including a real IP address 91, a transport protocol designation 92, a port number identification 93 and an Application layer protocol designation 94. The associations are provided at the time the virtual destination address is registered in the virtual network. For example, the entry 80 for the virtual IP address 10.0.0.12 (90) includes the association of a real IP address 152.70.0.1 (91), an associated Transport layer protocol designation, UDP 92, an associated port number, 6789 (93), and an Application layer protocol designation of ‘none’ 94.” Caronni II at 5:10–29.</p>

Exhibit C-29

Claims	VMware Systems
	<p>“FIG. 4 depicts the sequence of steps followed by the illustrative embodiment of the present invention to process the received encapsulated message. The sequence begins when a process associated with a virtual IP address registers with the virtual address resolution facility 24 (step 140). The registration is stored in a VARP lookup table accessible over the network. Subsequently, a message is received at the electronic device which bears a MAC/network interface address of the network interface of the electronic device with the designated real IP address listed in the virtual address registration (Step 142). The MAC address header 117 is stripped off at the Link layer and the message and appended headers are passed up to the Network layer 114(Step 144). The Network layer 114 strips off the IP header 115 and identifies the Transport Protocol header 113 underneath. The Network layer passes the message and remaining appended header to the Transport layer 112 (Step 146). The Transport layer 112 strips off the Transport header and then passes the message to the Application layer via the port address of the interface identified in the VARP registration. (Step 148). The receiving process determines whether it is the virtual destination address the message is addressed to (Step 149) and acts accordingly. If the receiving process that is examining the message is the destination address, the process examines the data (Step 150). Alternatively, if the message is not intended for the process executing on the edge device, the process re-routes the message and sends it back down the protocol model stack, on to the local network operating behind the edge device, and on to its intended destination by performing traditional VARP and ARP lookups. Alternatively forwarding the message may be accomplished by reinserting the packet into the IP layer of the networking stack.” Caronni II at 7:33–67.</p> <p><i>See also</i> Caronni II at Claim 1, FIGS. 2-4.</p> <p>“FIG. 3 is a data flow diagram illustrating the registration of virtual network environment parameters. The VNE framework 200 is a software module that processes transactions between the applications and the operating system. The VNE parameters are registered with the VNE framework 200 at the time the application is started. The VNE parameters include the application IP address, the virtual</p>

Exhibit C-29

Claims	VMware Systems
	<p>network subnet, and the global virtual address subnet. At step 250, the registration harness 220 supplies the application IP address, virtual subnet, and the global virtual address subnet for a processx to the VNE framework 200. The VNE framework 200 then records the IP address, virtual subnet, and the global virtual address subnet for the processx at step 252. The processx can then spawn additional processes (or create additional objects) at step 254. the new processy inherits the IP address, virtual subnet, and global virtual address subnet from processx. At step 256, the registration harness 220 launches the application related to processy.” Hipp at 3:52-4:2.</p> <p>“Virtualization of network identity is achieved by assigning a unique virtual IP address and virtual hostname to a group of processes that make up the application instance which the instance keeps throughout its execution. This virtual network identity stays with the application instance regardless of which node the application is running on. The framework, in essence, provides a mechanism to create this virtual network identity (VNI) around the application using the virtual network parameters assigned to it. In one embodiment, the virtual network parameters include an IP address and hostname. The framework 200 ensures that the application's instance uses the virtual network parameters, transparently, so that it can be moved across machines, without modifications to the application.” Hipp at 6:1-15.</p> <p><i>See also</i> Hipp at 6:16-60, FIG. 3.</p> <p>To the extent that VMware Systems does not disclose any elements of this claim limitation, it would have been obvious to a person having ordinary skill in the art to combine, with the teachings of VMware Systems , those elements, as known to them or as disclosed by other prior art identified in the cover pleading to Defendant’s Invalidity Contentions.</p>
<p>[30.4] distribute a virtual network address to the device when the device is registered in the virtual network, the device being</p>	<p>VMware Systems discloses or renders obvious the register module further configured to: distribute a virtual network address to the device when the device is registered in the</p>

Exhibit C-29

Claims	VMware Systems
<p>identified to other devices in the virtual network by the virtual network address; and</p>	<p>virtual network, the device being identified to other devices in the virtual network by the virtual network address. For example:</p> <p>“System resources are dynamically allocated to any operating system based on need, providing mainframe-class capacity utilization and control of server resources.” (VMware ESX Server Specifications)</p> <p>“Provision Servers Rapidly: Pre-configured virtual machine servers can be built once quickly and deployed anywhere immediately.” (VMware GSX Server Specifications)</p> <p>“By implementing a virtual infrastructure, IT organizations can provision new services and change the amount of resources dedicated to a software service simply by interacting with a management console.” (VMware Enterprises Continue to Embrace VMware Virtual Infrastructure)</p> <p>To the extent this reference does not expressly or inherently disclose this limitation, it would have been obvious in view of the background knowledge of a person of ordinary skill in the art. This reference further renders the limitation obvious in light of other references, such as Mehta, Caronni I, Caronni II and/or Hipp. Mehta, Caronni I, Caronni II and Hipp concern similar subject matter as this reference, namely, virtual networks. As set forth in the accompanying contentions, distributing virtual network addresses was a well-known and commonly-used technique. Modifying this reference with Mehta’s, Caronni I’s, Caronni II’s, and/or Hipp’s teachings would have only required simple substitution, would not have required extensive experimentation, and would have yielded predictable results. A person of ordinary skill in the art would have further understood how to modify the disclosures of this reference with those of Mehta, Caronni I, Caronni II, or Hipp for example, to incorporate this limitation. <i>See, e.g.,</i></p> <p>“Methods and systems for providing two-way initiated, bi-directional communication with wireless devices using connection-based or connection-less</p>

Exhibit C-29

Claims	VMware Systems
	<p>protocols, such as, for example, TCP/IP and UDP/IP, are provided. Example embodiments provide an Address Management Proxy System (“AMPS”), which enables devices and systems connected to a public internet, such as the Internet, to initiate communication with and to send data to wireless devices connected to a private wireless network, without exposing the non-routable private addresses of these wireless devices. The AMPS allocates a public (routable) network address for temporarily use by a requesting device on a public network to communicate with a wireless device on a wireless network. In one embodiment, a pool of public addresses, for example, public IP addresses, is maintained by the AMPS and allocated dynamically to wireless network devices as required. In one embodiment, the AMPS comprises one or more modified DNS/API servers, one or more Address Proxy/Routers, an Address Management Data Server, one or more data repositories, and optionally a load balancer. The AMPS DNS/API server receives a request from a device on a public network for a particular wireless device, and returns an appropriate temporary public address, which is internally mapped to the private address of the wireless device. The public address is then usable by the device on the public network to send data to the wireless device.” Mehta at Abstract.</p> <p>“To insure a greater degree of security, according to one embodiment, the AMPS maintains a Time to Live (TTL) parameter with each address mapping. In this way, once the TTL value indicates that the mapping has expired, the AMPS can destroy the mapping, and also any connection. Further, in some embodiments, the AMPS also puts a timestamp in its own mapping tables. After some timeout period based upon the timestamp, the AMPS can destroy the mapping, thereby forcing a new mapping to be initiated on a periodic basis.” Mehta at [0015].</p> <p>“In existing systems, data communication (communication on a data channel) between a wireless device that is connected to a private wireless network and a wired device connected to a public wired network (e.g., the Internet) can only be initiated by the wireless device. Some carriers have assigned fixed public IP addresses to wireless devices; however, the wireless devices need to then have</p>

Exhibit C-29

Claims	VMware Systems
	<p>client programs (e.g., a UDP stack) capability of receiving and handling the incoming communication packets. Moreover, these wireless devices are then part of a public wireless network and not a private wireless network. Since public IP addresses are becoming a scarcer commodity and currently expensive to service a network of multi-million devices, carriers cannot in a practical sense count on having a fixed public IP address for each device on its network. (Although movement to IPv6 will allow more addresses, the current IPv4 definitions are limited and the potential number of wireless users subscribing to larger carriers is very high. In some regions of the world, the current public address scheme is even more limited.) Further, such addressing capability would expose each device to further security risks, because each such device is part of a publicly accessible network and it becomes more difficult to implement and enforce security measures. Thus, assigning private IP addresses to devices is preferred in existing wireless networks over assigning fixed public IP addresses to devices. When wireless networks are private, the locations (addresses) of the wireless devices are intentionally hidden from public view by a carrier system's (or, as referred to in some countries, operator's) infrastructure.” Mehta at [0025].</p> <p>“FIG. 3 is an example block diagram of components of an example Address Management Proxy System. In one embodiment, the Address Management Proxy System (AMPS) comprises one or more modified DNS/ API servers 302, one or more Address Proxy/Routers 305, an Address Management Data Server 303 which manages a database or other repositories such as Address Management Data Repository 304, and optionally a load balancer 301. The DNS'/API servers 302 are either individually connected to a public network 310 or are connected to the load balancer 301 which is in turn connected to public network 310. Similarly, each Address Proxy/Router 305 is also connected to the public network 310, via the routable (public) address to which data from the external network 310 is sent that is destined for the wireless devices. The DNS'/API servers 302 are either modified implementations of a DNS server to add functionality necessary to communicate with wireless devices, or are servers that implement one or more specialized APIs, as will be described further below. The DNS'/API servers 302 use the Address</p>

Exhibit C-29

Claims	VMware Systems
	<p>Management Data Server 303 to assist in mapping a unique identifier (e.g., string name) for a wireless device to a public address on public network 310. The pool of public addresses is also maintained by the Address Management Data Serve 303 and Data Repository 304. The Address Management Data server 303 and Address Management Data Repository 304 may be implemented using existing database technology, for example, ODBC technology or, may be implemented as a structure such as a simple text file. One skilled in the art will recognize that any embodiment for storing a set of tables, data, lists, or mappings can be used. Each Address Proxy/Router 305 also uses the Address Management Data Server 303 or equivalent to create and update a series of routing tables that are used to assign public addresses to wireless devices as needed and to update the various mappings between public addresses and the non-routable (private) addresses of the wireless devices. The tables and mappings that are maintained on behalf of the DNS/API servers 302 and the Address Proxy/Routers 305 by the Address Management Data Server 303 are described below with reference to FIG. 6.” Mehta at [0032].</p> <p>“Although the techniques of the AMPS are generally applicable to any a wired device communicating with a wireless device, the phrase “public network” (or “wired network”) is used generally to imply any type of internetworked environment including a public network or a backbone that is somewhere down the line connected to one or more private or public networks. In addition, although the examples described herein often refer to the Internet, one skilled in the art will recognize that the concepts and inventions described are applicable to other forms and embodiments of internetworking, including, for example ATM type networks. Thus, techniques of the present invention can also be used by one device on a first wireless network to communicate with another wireless device on a second network—each device ends up communicating with the Address Proxy/Router of the other network. This scenario is feasible because each wireless network (or its carrier infrastructure) is connected to a proxy/router that is also connected (via a public address) to a public network. In addition, although a public network is sometimes also referred to herein as a wired network, one skilled in the art will recognize that any network that exposes routable (public) addresses may be</p>

Exhibit C-29

Claims	VMware Systems
	<p>implied. Thus, a wireless network with unique public (and routable) address can also employ techniques of the present invention to perform bi-directional communication. Also, one skilled in the art will recognize that terms such as wireless device, phone, handheld, etc., are used interchangeably to indicate any type of wireless device that is capable of operating with the AMPS. In addition, terms may have alternate spellings which may or may not be explicitly mentioned, and one skilled in the art will recognize that all such variations of terms are intended to be included.” Mehta at [0033].</p> <p>“Example embodiments described herein provide applications, tools, data structures and other support to implement private to public address mappings over one or more wired and wireless networks to be used for bi-directional communication. One skilled in the art will recognize that other embodiments of the methods and systems of the present invention may be used for many other purposes, including to push information and/or data or code from a public network such as the Internet to a wireless device. In addition, although this description primarily refers to “data” as being sent via the networks, one skilled in the art will recognize that all types of data can be communicated using the techniques described herein including, but not limited to, text, graphics, audio, and video.” Mehta at [0034].</p> <p>“There are several implementation approaches to the components of the Address Management Proxy System, three of which are described herein. One skilled in the art will recognize that various other approaches and combinations are possible. All three approaches allocate a public (routable) network address for temporary use by a wired device to communicate with a wireless device. In one embodiment, a pool of public addresses, for example, public IP addresses, is maintained and allocated dynamically to wireless network devices as required. For example, a typical Class B Internet network address block allows for approximately 65,000 simultaneous connections to wireless devices. Although this number may seem large at first</p>

Exhibit C-29

Claims	VMware Systems
	<p>blush, when one considers the number of cell phones and handsets, for example, connected to a carrier, this number can be quite limiting.” Mehta at [0039].</p> <p>“The unique ID table 610 maps unique string names of wireless devices to the private wireless network addresses that have been assigned typically by a carrier's infrastructure. In some embodiments, the carrier infrastructure dynamically allocates, using methods similar to a DHCP protocol, a private wireless network address when the wireless device registers itself with the carrier infrastructure upon being powered on. Thus, the unique ID table 610 may be sparsely filled or entries created dynamically and then deleted dynamically as devices register and unregister with the carrier infrastructure system.” Mehta at [0053].</p> <p>“In one embodiment, a timestamp of the mapping between a public network address and a private address is also noted in table 630. After a defined time out, based upon this timestamp, the Address Management Data Server sends a request to the Address Proxy/Router associated with the mapping to unmap the public-to-private address mapping and updates the mapping table 630, thereby returning the associated public address to the pool of unused public network addresses.” Mehta at [0056].</p> <p>“FIG. 7 is an example flow diagram of an example routine provided by a DNS/API server of the Address Management Proxy System to return a public address that corresponds to a designated unique identifier. In essence, this routine implements a DNS query or DNS-like query capability for the AMPS using a modified GetHostByName interface or a specialized API, for example GetProxyIP, as described with reference to FIG. 5. In summary, the routine dynamically allocates an appropriate Address Proxy/Router machine to associate with the wireless device and returns the public address of that machine (along with a TTL parameter and potentially other parameters). Specifically, in step 701, the routine determines the private (non-routable) address of the wireless device designated by a string parameter passed as input to the routine. For example, the string parameter may use fields such as “uniqueID.hostname.domain.tld,” which specifies a typical hierarchy</p>

Exhibit C-29

Claims	VMware Systems
	<p>of person/service on a machine named “hostname” on a domain such as a company's network on a top level domain such as “org.” “com,” “edu,” etc. One skilled in the art will recognize that many other string parameter designations could be used. One mechanism for implementing this routine is to request information from the Address Management Data Repository. In one embodiment, the data repository stores a table that maps unique IDs to private network addresses (see Table 610 in FIG. 6). Preferably, any mechanism that is used by the AMPS stores this data in a secure manner in order to keep the wireless network addresses private. In step 702, the routine retrieves the public network address that corresponds to the private wireless network address of the designated device if one has already been assigned by the AMPS to that device and is still valid. In one embodiment, the data repository stores this mapping information between private wireless network address and public network address (see for example table 630 in FIG. 6). If a public network address has not already been assigned or is not valid, then the routine causes a new public network address to be allocated and that new public address is associated with the private wireless network address. Appropriate tables in the data repository are then updated. In step 703, the routine determines the Address Proxy/Router machine that is associated with the assigned public network address (for example using table 620 in FIG. 6). In step 704, the routine sends a request to the determined proxy/router machine to update its routing tables to map the determined public network address to the private wireless network address. In step 705, the routine updates information in the data repository to include any other connection related information (e.g., field 634 in Table 630 in FIG. 6) and indicates a Time to Live (TTL) parameter for the public-private address association (e.g., field 633 in Table 630 in FIG. 6). Once all of the tables have been updated in both the proxy/router and in the data repository, the DNS'/API server returns the determined public network address of the associated Address Proxy/Router machine. As described earlier, the public address may be a (hostname, port) pair when a port-based implementation is used.” Mehta at [0057].</p>

Exhibit C-29

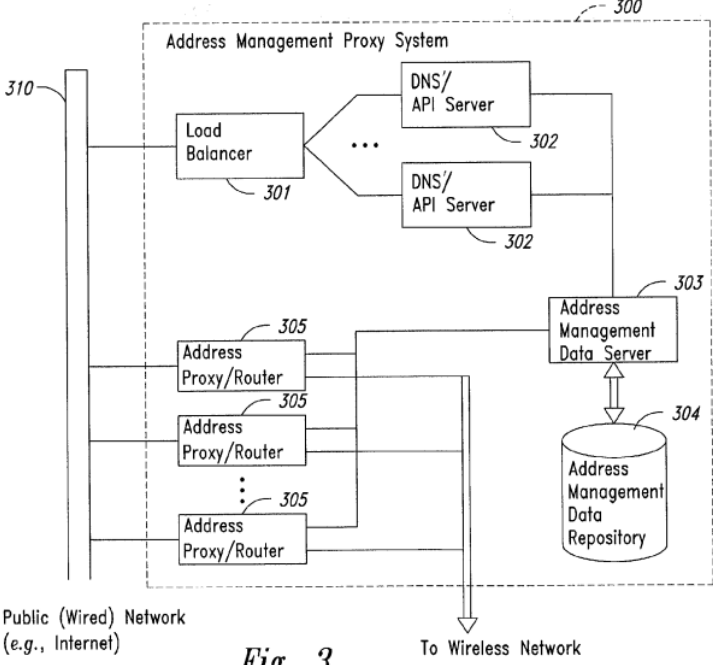
Claims	VMware Systems
	 <p>The diagram, labeled Fig. 3, illustrates an Address Management Proxy System (300). On the left, a vertical bar labeled 310 represents the Public (Wired) Network (e.g., Internet). This network is connected to a Load Balancer (301) which distributes traffic to multiple DNS/API Servers (302). Below the Load Balancer, there are several Address Proxy/Routers (305). The DNS/API Servers (302) are connected to an Address Management Data Server (303). The Address Proxy/Routers (305) are also connected to the Address Management Data Server (303) and to an Address Management Data Repository (304), which is represented as a cylinder. The Address Management Data Server (303) has a bidirectional connection with the Address Management Data Repository (304). The system (300) is shown with an arrow pointing downwards to a Wireless Network.</p> <p style="text-align: center;"><i>Fig. 3</i></p> <p>Mehta at FIG. 3.</p> <p>“Fourth, the system provides address translation in a transparent manner. Since the Supernet is a private network constructed from the infrastructure of another network, the Supernet has its own internal addressing scheme, separate from the addressing scheme of the underlying public network. Thus, when a packet from a Supernet node is sent to another Supernet node, it travels through the public network. To do so, the Supernet performs address translation from the internal addressing scheme to the public addressing scheme and vice versa. By separating</p>

Exhibit C-29

Claims	VMware Systems
	<p>the addressing schemes, the Supernet creates a flexible delivery scheme that is easily changeable by network software or a system administrator. To reduce the complexity of Supernet nodes, system-level components of the Supernet perform this translation on behalf of the individual nodes so that it is transparent to the nodes. Another benefit of the Supernet's addressing is that it uses an IP-based internal addressing scheme so that preexisting programs require little modification to run within a Supernet.” Caronni I at 6:7–25.</p> <p>“SASD 540 represents a Supernet: there is one instance of an SASD per Supernet, and it both authenticates nodes and authorizes nodes to join the Supernet. VARPD 548 has an associated component, VARPDB 551, into which it stores mappings of the internal Supernet addresses, known as a node IDs, to the network addresses recognized by the public-network infrastructure, known as the real addresses. The “node ID” may include the following: a Supernet ID (e.g., 0x123), reflecting a unique identifier of the Supernet, and a virtual address, comprising an IP address (e.g., 10.0.0.1). Although the virtual address is described in an IP address scheme, one skilled in the art will appreciate that the virtual address may be any other type addressing scheme, such as an e-mail address, IPX, or IPv6. Since the node ID includes a Supernet ID, a node will have more than one node ID when it communicates over more than one channel. The “real address” is an IP address (e.g., 10.0.0.2) that is globally unique and meaningful to the public-network infrastructure. In a Supernet, one VARPD runs on each machine, and it may play two roles. First, a VARPD may act as a server by storing all address mappings for a particular Supernet into its associated VARPDB. Second, regardless of its role as a server or not, each VARPD assists in address translation for the nodes on its machine. In this role, the VARPD stores into its associated VARPDB the address mappings for its nodes, and if it needs a mapping that it does not have, it will contact the VARPD that acts as the server for the given Supernet to obtain it. The VARPDB may also decide which virtual address to use in the translation. That is, the VARPDB may associate a virtual address with multiple real addresses or vice versa.” Caronni I at 7:3–33.</p>

Exhibit C-29

Claims	VMware Systems
	<p>“FIGS. 7A and 7B depict a flow chart of the steps performed when a node joins a Supernet. The first step performed is that the user invokes the SNlogin script and enters the Supernet name, their user ID, their password, and a requested virtual address (step 702). Of course, this information depends on the particular authentication mechanism used. Upon receiving this information, the SNlogin script performs a handshaking with SASD to authenticate this information. In this step, the user may request a particular virtual address to be used, or alternatively, the SASD may select one for them. Next, if any of the information in step 702 is not validated by SASD (step 704), processing ends. Otherwise, upon successful authentication, SASD creates an address mapping between a node ID and the real address (step 706). In this step, SASD concatenates the Supernet ID with the virtual address to create the node ID, obtains the real address of the SNlogin script by querying network services in a well-known manner, and then registers this information with the VARP that acts as the server for this Supernet. This VARP is identified in the configuration file.” Caronni I at 9:66-10:18.</p> <p>“Once inner IP layer 542 receives the packet, a Supernet ID is appended to a socket structure (step 804). The socket structure is modified so as to contain an extra data field for Supernet ID 626 and virtual source address 642. The addition of Supernet ID 626 and virtual address 642 in the socket structure enables the Supernet to communicate with nodes regardless of the delivery scheme used. When the process on node A opens a socket to transmit the packet to inner IP layer 540, the corresponding Supernet ID 626 and virtual source address 642 for that process is included in the socket request.</p> <p>The packet and Supernet ID are then transmitted to the SNSL layer using the modified socket structure (step 806). The SNSL layer then accesses the VARPDB to obtain the address mapping between virtual source node address 642 and the source real address 614 as well as the virtual destination node address 644 and the destination real address 616 (step 808). If they are not contained in the VARPDB because this is the first time a packet has been sent from this node or sent to this destination, the VARPDB accesses the local VARP to obtain the mapping. When</p>

Exhibit C-29

Claims	VMware Systems
	<p>contacted, the VARPDB on the local machine contacts the VARPDB that acts as the server for the Supernet to obtain the appropriate address mapping. Since the VARPDB maintains all real IP addresses, a remote node may securely communicate with another remote node without reverification.” Caronni I at 11:37–61.</p> <p><i>See also</i> Caronni I at FIGS. 7A and 7B.</p> <p>“Unfortunately, when the virtual destination address is located on a physical device on the interior of a network which is running a proxy server, firewall, or other packet filtering mechanism, messages that have been sent to a virtual destination address have difficulty getting all the way to their target. The term “interior of a network” refers to devices which are not able to directly access another network without first going through another device on their own network. For example, most local area networks (LANs) access the Internet through a proxy server. Devices other than the proxy server are said to be on the interior of the LAN. The proxy server is referred to as an “edge device” because it is able to directly contact another network without using an intermediary device. “Packet filtering” refers to the filtering of incoming messages or packets by an edge device or process on an edge device so that not all of the packets are permitted to proceed to their destination, they are “filtered out”. If the electronic device that is filtering incoming packets, is under the control of the party executing the process associated with the virtual destination address, the device may be configured to allow the packets through to the end destination. However, in many situations, the edge device is not configurable by anyone without system administration privileges. Similarly, if the edge device is a device performing Network Address Translation (a “NAT box”), the NAT box rewrites all outgoing packets from an end user in the interior of the network to make them look like they came directly from the NAT box, and remembers that any traffic coming back from the particular destination address must be mapped back to the originating internal device. Consequently, the responding devices think they are responding to the sending device when they are actually responding to an edge device. In such a case, the packets intended for the</p>

Exhibit C-29

Claims	VMware Systems
	<p>virtual destination address on the interior of the physical network may be dropped and not reach their intended destination.” Caronni II at 1:60–2:27.</p> <p>“...sending said formatted message, augmented by said determined MAC address destination, said real IP address destination and said transport protocol header, from a virtual IP address to the real IP address destination indicated in said virtual address destination registration.” Caronni II at 9:20–24.</p> <p>“...a process associated with the sending virtual IP address is located on an electronic device outside a firewall and said virtual address destination is located inside the firewall.” Caronni II at 9:31–32.</p> <p><i>See also</i> Caronni II at 2:31-3:24, 6:32-54, FIGS. 1-5.</p> <p>“The Virtual Network Environment (VNE) of the present invention is defined by a collection of IP addresses. An application running within one VNE can communicate with another application in the same VNE. However, an application in one VNE cannot communicate with an application in another VNE (unless expressly permitted). These and many other attendant advantages of the present invention will be understood upon reading the following detailed description in conjunction with the drawings.” Hipp at 2:8-16.</p> <p>“The VNE is specified at application run time. The VNE is transparent to the application and does not require any modifications to the application. The VNE is defined by subnet of addresses contained within the VNE. For example, all applications within the subnet 10.10.2.0 comprise a VNE. The subnet/netmask specifying such a VNE would be 10.10.2.0/255.255.255.0 and would include the addresses 10.10.2.0 through 10.10.2.255. In this example, an application with IP address 10.10.2.2 would be able to communicate with an application at address 10.10.2.60, but not at 10.10.0.1. Although using a subnet/netmask to specify the VNE is described herein for illustrative purposes, it is to be understood that other</p>

Exhibit C-29

Claims	VMware Systems
	<p>methods may be used to accomplish the same mechanism (e.g. an access control list).” Hipp at 3:16-30.</p> <p>“Virtualization of network identity is achieved by assigning a unique virtual IP address and virtual hostname to a group of processes that make up the application instance which the instance keeps throughout its execution. This virtual network identity stays with the application instance regardless of which node the application is running on. The framework, in essence, provides a mechanism to create this virtual network identity (VNI) around the application using the virtual network parameters assigned to it. In one embodiment, the virtual network parameters include an IP address and hostname. The framework 200 ensures that the application's instance uses the virtual network parameters, transparently, so that it can be moved across machines, without modifications to the application.” Hipp at 6:1-15.</p> <p><i>See also</i> Hipp at Figures 1–7.</p> <p>To the extent that VMware Systems does not disclose any elements of this claim limitation, it would have been obvious to a person having ordinary skill in the art to combine, with the teachings of VMware Systems, those elements, as known to them or as disclosed by other prior art identified in the cover pleading to Defendant’s Invalidation Contentions.</p>
<p>[30.5] a DNS server for the virtual network, the DNS server configured to receive a DNS request from a first device in the virtual network, and return a network address associated with a network route director, a private network address associated with a second device in the virtual network, and a virtual</p>	<p>VMware Systems discloses or renders obvious a DNS server for the virtual network, the DNS server configured to receive a DNS request from a first device in the virtual network, and return a network address associated with a network route director, a private network address associated with a second device in the virtual network, and a virtual network address associated with the second device.</p> <p>To the extent this reference does not expressly or inherently disclose this limitation, it would have been obvious in view of the background knowledge of a person of ordinary</p>

Exhibit C-29

Claims	VMware Systems
<p>network address associated with the second device.</p>	<p>skill in the art. This reference further renders the limitation obvious in light of other references, such as Mehta, RFC 1383, Caronni I, Caronni II, and/or Hipp. Mehta, RFC 1383, Caronni I, Caronni II, and/or Hipp concern similar subject matter as this reference, namely, virtual networks. As set forth in the accompanying contentions, use of a server such as a DNS server that returns network addresses was a well-known and commonly-used technique. Modifying this reference with Mehta's, RFC 1383's, Caronni I's, Caronni II's, and/or Hipp's teachings would have only required simple substitution, would not have required extensive experimentation, and would have yielded predictable results. A person of ordinary skill in the art would have further understood how to modify the disclosures of this reference with those of Mehta, RFC 1383, Caronni I, Caronni II, or Hipp for example, to incorporate this limitation. <i>See, e.g.,</i></p> <p>“A network that supports IP can be connected to another TCP/IP-based or UDP/IP-based internet or the Internet, by providing a router which forwards data to another router or host machine based upon an IP address. The router (or routing server/service) typically contains a routing table, which determines to which machine (and optionally to which port) to send a datagram, given a destination IP address. The IP address uniquely identifies a router/host machine, and, in a typical TCP/IP network, can be mapped to a string name that identifies, for example, a particular machine as part of a larger domain. (Although referred to herein often as a TCP/IP-based network, one skilled in the art will recognize that the network may also be UDP-based (connectionless), and may support another session management system.)” Mehta at [0008].</p> <p>“In one embodiment, the AMPS comprises one or more modified DNS/API servers, one or more Address Proxy/Routers, an Address Management Data Server, one or more Address Management Data Repositories, and optionally a load balancer. The AMPS DNS/API server receives a request from a device on a public network for a particular wireless device, and returns an appropriate temporary public address, which is internally mapped to the private address of the wireless</p>

Exhibit C-29

Claims	VMware Systems
	<p>device. The public address is then usable by the device on the external public network to send data to the wireless device. The temporary public address is, for example, an address associated with one of the Address Proxy/Routers, which are connected to the external public network and have access to the private addresses on the private wireless network. In some cases, the device on the public network that desires to send data to the wireless device uses a connection-based protocol, such as TCP/IP to send data. In other cases the device uses a connection-less protocol, such as UDP (UDP/IP) to send the data.” Mehta at [0012].</p> <p>“Wireless systems on private networks use techniques similar to Network Address Translation (NAT) technology to send data from a wireless device to the public networking world. In a typical carrier infrastructure that uses a private network, a wireless device “registers” itself with the carrier infrastructure when it powers on (or in other circumstances, when the device attempts to initiate data services). The carrier dynamically assigns the wireless device a private non-routable address by means of DHCP (or DHCP-like) server. The information on the mapping of the transient private IP address to the device public IP address is stored in an internal carrier database and managed by carrier services such as a RADIUS server.” Mehta at [0026].</p> <p>“The Address Management Proxy System achieves two way initiated bi-directional communication by implementing a modified DNS server and serving as a proxy/router for devices on the private wireless network as they interface to the public wired internetworking world. In summary, a pool of public addresses is maintained and dynamically distributed among active wireless devices as needed by the AMPS. FIG. 2 is a block diagram of an example Address Management Proxy System used in bi-directional communication with a wireless device. The term “bi-directional” as used herein means that data paths and communication can flow in either direction between two endpoint systems. FIG. 2 shows wired devices 201, 202, and 203 connected to public network 210. One skilled in the art will recognize that these devices could be connected to another private or public network, which is then connected by one or more wired devices to the public</p>

Exhibit C-29

Claims	VMware Systems
	<p>network 210, yet still achieve the functionality discussed here. Any such variation provides equivalent functionality and is explicitly contemplated and presumed to be part of the present invention. On the wireless side, the AMPS 220, acting in its capacity as a proxy (and router) for wireless devices, is shown both connected via wire to the public network 210 and via standard carrier infrastructure elements (not shown) to the wireless network 230. It is presumed that the reader has a working knowledge of the elements of a carrier's infrastructure and the basic mechanisms for routing and mechanisms converting packets from a wired network to a wireless network. These may use analog or digital technology and may require protocol conversions in order to send the physical data and transmit it, for example through a satellite, ultimately to the wireless device. Detailed background information on wireless technology and wireless routing mechanisms is described in Stallings, W., Wireless Communications and Networks, Prentice Hall, N.J., 2002, which is herein incorporated by reference in its entirety. In FIG. 2, wireless device 240 is shown connected via various wireless elements (not shown) to the wireless network 230.” Mehta at [0030].</p> <p>“FIG. 6 is an example block diagram of some of the Address Management Proxy System data repository tables used to support routines of the DNS/API servers and the Address Proxy/Routers. In one embodiment, the Address Management Data Repository comprises three tables: a unique identifier (unique ID) to private address table 610, a public-to-private address table 630, and a public address to proxy/router machine table 620. Although three tables are shown, one skilled in the art will recognize that these tables could contain other data and may be organized differently, including a different number of tables and with different columns or fields. In addition, any technique for storing a table or list of data may be used. To support embodiments that map a host address plus a port designator to a wireless device, the tables are correspondingly modified.” Mehta at [0052].</p> <p>“The unique ID table 610 maps unique string names of wireless devices to the private wireless network addresses that have been assigned typically by a carrier's infrastructure. In some embodiments, the carrier infrastructure dynamically</p>

Exhibit C-29

Claims	VMware Systems
	<p>allocates, using methods similar to a DHCP protocol, a private wireless network address when the wireless device registers itself with the carrier infrastructure upon being powered on. Thus, the unique ID table 610 may be sparsely filled or entries created dynamically and then deleted dynamically as devices register and unregister with the carrier infrastructure system.” Mehta at [0053].</p> <p>The public-to-private address table 630 comprises several fields/columns including a public network address 631, a private (wireless) network address 602, a flag 632 that specifies whether the public address stored in field 631 is free or is already used, a Time to Live (TTL) parameter 633, and other connection data 634. In one embodiment, the DNS/API servers of the AMPS query table 630 to determine a public network address that corresponds to a designated private wireless network address or to allocate an unused public network address (as indicated in field 632) and map the determined unused public address to a private network address stored in field 602. Mehta at [0054].</p> <p>Public address to proxy/router machine table 620 comprises a public network address field 631 and an indication of a functioning proxy/router machine 621. By maintaining such a mapping, the AMPS is able to substitute proxy/router machines for other proxy/router machines to provide a higher degree of robustness. Each proxy/router machine has a preconfigured set of public network addresses, such as are typically configured by network cards inserted into the proxy/router machine. These address are allocated in a standard fashion through prior purchase or obtaining from an address authorizing authority, currently the Internet Corporation for Assigned Names and Numbers (ICANN). When a machine is inserted for use into the AMPS, indications of these addresses are stored in field 631. Mehta at [0055].</p> <p>“In one embodiment, a timestamp of the mapping between a public network address and a private address is also noted in table 630. After a defined time out, based upon this timestamp, the Address Management Data Server sends a request to the Address Proxy/Router associated with the mapping to unmap the public-to-</p>

Exhibit C-29

Claims	VMware Systems
	<p>private address mapping and updates the mapping table 630, thereby returning the associated public address to the pool of unused public network addresses.” Mehta at [0056].</p> <p>“FIG. 7 is an example flow diagram of an example routine provided by a DNS/API server of the Address Management Proxy System to return a public address that corresponds to a designated unique identifier. In essence, this routine implements a DNS query or DNS-like query capability for the AMPS using a modified GetHostByName interface or a specialized API, for example GetProxyIP, as described with reference to FIG. 5. In summary, the routine dynamically allocates an appropriate Address Proxy/Router machine to associate with the wireless device and returns the public address of that machine (along with a TTL parameter and potentially other parameters). Specifically, in step 701, the routine determines the private (non-routable) address of the wireless device designated by a string parameter passed as input to the routine. For example, the string parameter may use fields such as “uniqueID.hostname.domain.tld,” which specifies a typical hierarchy of person/service on a machine named “hostname” on a domain such as a company's network on a top level domain such as “org.” “com,” “edu,” etc. One skilled in the art will recognize that many other string parameter designations could be used. One mechanism for implementing this routine is to request information from the Address Management Data Repository. In one embodiment, the data repository stores a table that maps unique IDs to private network addresses (see Table 610 in FIG. 6). Preferably, any mechanism that is used by the AMPS stores this data in a secure manner in order to keep the wireless network addresses private. In step 702, the routine retrieves the public network address that corresponds to the private wireless network address of the designated device if one has already been assigned by the AMPS to that device and is still valid. In one embodiment, the data repository stores this mapping information between private wireless network address and public network address (see for example table 630 in FIG. 6). If a public network address has not already been assigned or is not valid, then the routine causes a new public network address to be allocated and that new public address is associated with the private wireless network address. Appropriate tables</p>

Exhibit C-29

Claims	VMware Systems
	<p>in the data repository are then updated. In step 703, the routine determines the Address Proxy/Router machine that is associated with the assigned public network address (for example using table 620 in FIG. 6). In step 704, the routine sends a request to the determined proxy/router machine to update its routing tables to map the determined public network address to the private wireless network address. In step 705, the routine updates information in the data repository to include any other connection related information (e.g., field 634 in Table 630 in FIG. 6) and indicates a Time to Live (TTL) parameter for the public-private address association (e.g., field 633 in Table 630 in FIG. 6). Once all of the tables have been updated in both the proxy/router and in the data repository, the DNS'/API server returns the determined public network address of the associated Address Proxy/Router machine. As described earlier, the public address may be a (hostname, port) pair when a port-based implementation is used.” Mehta at [0057].</p>

Exhibit C-29

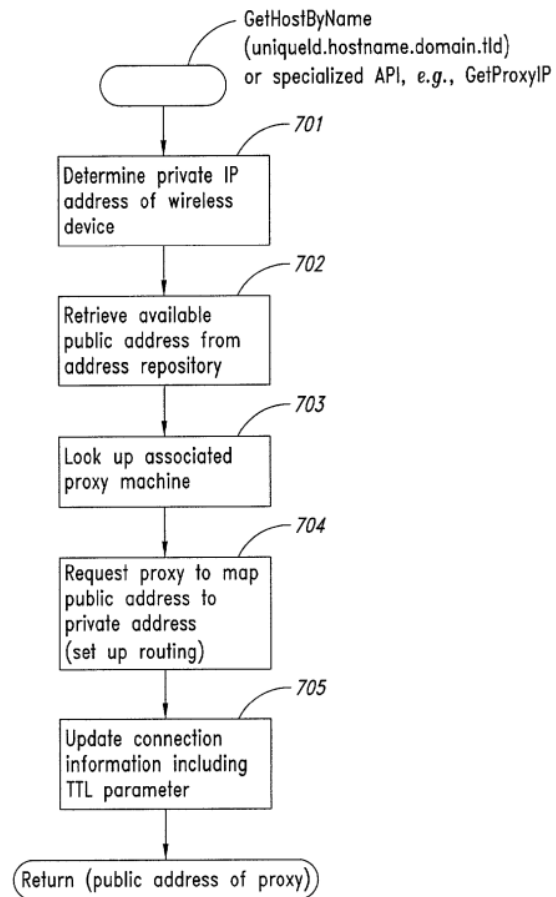


Fig. 7

Exhibit C-29

Claims	VMware Systems
	<p>Mehta at FIG. 7.</p> <p><i>See also</i> Mehta at [0004], [0010], [0032], [0036]-[0038], [0048], [0058]-[0059].</p> <p>“1. Routing, scaling and hierarchies</p> <p>Several recent studies have outlined the risk of ‘routing explosion’ in the current Internet: there are already more than 5000 networks announced in the NSFNET routing tables, more than 7000 in the EBONE routing tables. As these numbers are growing, several problems occur:</p> <ul style="list-style-type: none"> * The size of the routing tables grows linearly with the number of connected networks; handling this larger tables requires more resources in all ‘intelligent’ routers, in particular in all ‘transit’ and ‘external’ routers that cannot rely on default routes. * The volume of information carried by the route exchange protocols such as BGP grows with the number of networks, using more network resources and making the reaction to routing events slower. * Explicit administrative decisions have to be exercised by all transit networks administrators which want to implement ‘routing policies’ for each and every additional ‘multi-homed’ network. <p>The current ‘textbook’ solution to the routing explosion problem is to use “hierarchical routing” based on hierarchical addresses. This is largely documented in routing protocols such as IDRP, and is one of the rationales for deploying the CIDR [3] addressing structure in the Internet. This textbook solution, while often perfectly adequate, as a number of inconveniences, particularly in the presence of ‘multihomed stubs’, e.g., customer networks that are connected to more than one service providers.</p>

Exhibit C-29

Claims	VMware Systems
	<p>The current proposal presents a scheme that allows for simple routing. It is complementary with the classic ‘hierarchical routing’ approach, but provides an easy to implement and low cost solution for "multi-homed" domains. The solution is a generalization of the ‘MX record’ scheme currently used for mail routing.” RFC 1383 at 1-2.</p> <p>“3.1. Loops and relays</p> <p>In the introduction to DNS-IP routing, we mentioned that the packets would be directed towards the access gateway I1 or I2 by means of ‘source routing’ or ‘tunnelling’. This is not, stricto sensu, necessary. One could imagine that the packet would simply be routed ‘as if it was directed towards I1 or I2’. The next relay would, in turn, also access the DNS to get routing information and forward the packet.</p> <p>Such a strategy would have the advantage of leaving the header untouched and of letting the transit nodes choose the best routing towards the destination, based on their knowledge of the reachability status. It would however have two important disadvantages:</p> <ul style="list-style-type: none"> - It would oblige all intermediate relays to access the DNS, - It would oblige all these relays to exploit consistently the DNS information. <p>Obliging all intermediate gateways to access the DNS is impractical in the short term: it would mean that we would have to update each and every transit relay before deploying the scheme. It could also have an important performance impact: the ‘working set’ of transit relays is typical much wider than that of stub gateways, and the argument presented previously on the efficiency of caches may not apply.</p>

Exhibit C-29

Claims	VMware Systems
	<p>This would perhaps remain impractical even in the long term, as it the volume of DNS traffic could well become excessive.</p> <p>The second argument would apply even if the performance problem had been solved. Suppose that several RX records are registered for a given destination, such as I1 and I2 for Dx in our example, and that a ‘hop by hop routing’ strategy is used. There would be a fair risk that some relays would choose to route the packet towards I1 and some others towards I2, resulting in inefficient routing and the possibility of loops.</p> <p>In order to ensure coherency, we propose that all routing decisions be made at the source, or by one of the first relays near the source.” RFC 1383 at 4-5.</p> <p>“3.4. Choosing a gateway</p> <p>A simplification to the previous problem would be to allow only one RX record per destination, thus guaranteeing consistent decisions in the network. This would however have a number of draw-backs. A single access point would be a single point of failure, and would be connected to only one transit network thus keeping the ‘customer locking’ effect of hierarchical routing.</p> <p>We propose that the RX records have a structure parallel to that of MX records, i.e., that they carry associated with each gateway address a preference identifier. The source host, when making the routing decision based on RX records, should do the following:</p> <ul style="list-style-type: none"> - List all possible gateways, - Prune all gateways in the list which are known as ‘unreachable’ from the local site,

Exhibit C-29

Claims	VMware Systems
	<ul style="list-style-type: none">- If the local host is present in the list with a preference index 'x', prune all gateways whose preference index are larger than 'x' or equal to 'x'.- Choose one of the gateway in the list. If the list is empty, consider the destination as unreachable. <p>Indeed, these evaluations should not be repeated for each and every packet. The routers should maintain a cache of the most frequently used destinations, in order to speed up the processing.” RFC 1383 at 6.</p> <p>“5.1. DNS record</p> <p>In a definitive scheme, it would be necessary to define a DNS record type and the corresponding 'RX' format. In order to deploy this scheme, we would then have to teach this new format to the domain name service software. While not very difficult to do, this would probably take a couple of month, and will not be used in the early experimentations, which will use the general purpose 'TXT' record.</p> <p>This record is designed for easy general purpose extensions in the DNS, and its content is a text string. The RX record will contain three fields:</p> <ul style="list-style-type: none">- A record identifier composed of the two characters 'RX'. This is used to disambiguate from other experimental uses of the 'TXT' record.- A cost indicator, encoded on up to 3 numerical digits. The corresponding positive integer value should be less that 256, in order to preserve future evolutions.- An IP address, encoded as a text string following the 'dot' notation.

Exhibit C-29

Claims	VMware Systems								
	<p>The three strings will be separated by a single comma. An example of record would thus be:</p> <table border="1" data-bbox="835 370 1780 477"> <thead> <tr> <th>domain</th> <th>type</th> <th>record</th> <th>value</th> </tr> </thead> <tbody> <tr> <td>*.27.32.192.in-addr.arpa</td> <td>IP</td> <td>TXT</td> <td>RX, 10, 10.0.0.7</td> </tr> </tbody> </table> <p>which means that for all hosts whose IP address starts by the three octets ‘192.32.27’ the IP host ‘10.0.0.7’ can be used as a gateway, and that the preference value is 10.” RFC 1383 at 11-12.</p> <p><i>See also</i> RFC 1383 at 9, 12-13.</p> <p>“To perform this functionality, D 1 108 utilizes a technique known as tunneling to ensure that the communication between itself and enterprise network 102 is secure in that it cannot be viewed by an interloper. ‘Tunneling’ refers to encapsulating one packet inside another when packets are transferred between two end points (e.g., D 1 108 and VPN software 109 running on firewall 106). The packets may be encrypted at their origin and decrypted at their destination. For example, FIG. 2A depicts a packet 200 with a source Internet protocol (IP) address 202, a destination IP address 204, and data 206. It should be appreciated that packet 200 contains other information not depicted, such as the source and destination port. As shown in FIG. 2B, the tunneling technique forms a new packet 208 out of packet 200 by encrypting it and adding both a new source IP address 210 and a new destination IP address 212. In this manner, the contents of the original packet (i.e., 202, 204, and 206) are not visible to any entity other than the destination. Referring back to FIG. 1, by using tunneling, remote device D 1 108 may communicate and utilize the resources of the enterprise network 102 in a secure manner.” Caronni I at 2:26–46.</p> <p>“SASD 540 represents a Supernet: there is one instance of an SASD per Supernet, and it both authenticates nodes and authorizes nodes to join the Supernet. VARPD</p>	domain	type	record	value	*.27.32.192.in-addr.arpa	IP	TXT	RX, 10, 10.0.0.7
domain	type	record	value						
*.27.32.192.in-addr.arpa	IP	TXT	RX, 10, 10.0.0.7						

Exhibit C-29

Claims	VMware Systems
	<p>548 has an associated component, VARPDB 551, into which it stores mappings of the internal Supernet addresses, known as a node IDs, to the network addresses recognized by the public-network infrastructure, known as the real addresses. The “node ID” may include the following: a Supernet ID (e.g., 0x123), reflecting a unique identifier of the Supernet, and a virtual address, comprising an IP address (e.g., 10.0.0.1). Although the virtual address is described in an IP address scheme, one skilled in the art will appreciate that the virtual address may be any other type addressing scheme, such as an e-mail address, IPX, or IPv6. Since the node ID includes a Supernet ID, a node will have more than one node ID when it communicates over more than one channel. The ‘real address’ is an IP address (e.g., 10.0.0.2) that is globally unique and meaningful to the public-network infrastructure. In a Supernet, one VARPDB runs on each machine, and it may play two roles. First, a VARPDB may act as a server by storing all address mappings for a particular Supernet into its associated VARPD. Second, regardless of its role as a server or not, each VARPDB assists in address translation for the nodes on its machine. In this role, the VARPDB stores into its associated VARPD the address mappings for its nodes, and if it needs a mapping that it does not have, it will contact the VARPDB that acts as the server for the given Supernet to obtain it. The VARPD may also decide which virtual address to use in the translation. That is, the VARPD may associate a virtual address with multiple real addresses or vice versa.” Caronni I at 7:3–33.</p> <p>“SNSL 542 utilizes VARPD 536 to perform address translation. VARPD stores all of the address mappings encountered thus far by SNSL 542. If SNSL 542 requests a mapping that VARPD 536 does not have, VARPD communicates with the VARPD 526 on the local machine to obtain the mapping. VARPD 526 will then contact the VARPD that acts as the server for this particular Supernet to obtain it.” Caronni I at 9:47–54.</p> <p>“FIGS. 7A and 7B depict a flow chart of the steps performed when a node joins a Supernet. The first step performed is that the user invokes the SNlogin script and enters the Supernet name, their user ID, their password, and a requested virtual</p>

Exhibit C-29

Claims	VMware Systems
	<p>address (step 702). Of course, this information depends on the particular authentication mechanism used. Upon receiving this information, the SNlogin script performs a handshaking with SASD to authenticate this information. In this step, the user may request a particular virtual address to be used, or alternatively, the SASD may select one for them. Next, if any of the information in step 702 is not validated by SASD (step 704), processing ends. Otherwise, upon successful authentication, SASD creates an address mapping between a node ID and the real address (step 706). In this step, SASD concatenates the Supernet ID with the virtual address to create the node ID, obtains the real address of the SNlogin script by querying network services in a well-known manner, and then registers this information with the VARPDB that acts as the server for this Supernet. This VARPDB is identified in the configuration file.” Caronni I at 9:66-10:18.</p> <p>“The packet and Supernet ID are then transmitted to the SNSL layer using the modified socket structure (step 806). The SNSL layer then accesses the VARPDB to obtain the address mapping between virtual source node address 642 and the source real address 614 as well as the virtual destination node address 644 and the destination real address 616 (step 808). If they are not contained in the VARPDB because this is the first time a packet has been sent from this node or sent to this destination, the VARPDB accesses the local VARPDB to obtain the mapping. When contacted, the VARPDB on the local machine contacts the VARPDB that acts as the server for the Supernet to obtain the appropriate address mapping. Since the VARPDB maintains all real IP addresses, a remote node may securely communicate with another remote node without reverfication.” Caronni I at 11:47–61.</p> <p>“After obtaining the address mapping, the SNSL layer determines whether it has been configured to communicate over the appropriate channel for this packet (step 806). This configuration occurs when SNlogin runs, and if the SNSL has not been so configured, processing ends. Otherwise, SNSL obtains the channel key to be used for this channel (step 808). The SNSL maintains a local cache of keys and an indication of the channel to which each key is associated. Each channel key is time</p>

Exhibit C-29

Claims	VMware Systems
	<p>stamped to expire in ten seconds, although this time is configurable by the administrator. If there is a key located in the cache for this channel, SNSL obtains the key. Otherwise, SNSL accesses KMD which then locates the appropriate channel key from the appropriate KMC. After obtaining the key, the SNSL layer encrypts the packet using the appropriate encryption algorithm and the key previously obtained (step 810). When encrypting the packet, the virtual source node address 642, the virtual destination node address 644, and the data may be encrypted (addressing section 660), but the source and destination real addresses 614, 616 (delivery scheme section 670) are not, so that the real addresses can be used by the public network infrastructure to send the packet to its destination. By encrypting addressing scheme 660, the Supernet enables data to be transmitted securely and at the same time transparent from delivery scheme used.” Caronni I at 11:62-12:19.</p> <p>“Web client 1102 has a virtual address obtained from computer system 1106, as described in FIGS. 7A and 7B. Each time web client 1102 requests a packet from web server 1104 a, the client requests the virtual address of the web server 1104 a from computer system 1106. If web server 1104 a becomes overloaded (e.g., unable to handle more requests), the overloaded web server spawns new instances of the same web server (web servers 1104 b and 1104 c) and at the same time notifies the computer system 1106. The VARP server of computer 1106 then associates the new instances 104 b and 104 c of the web server with web server's 1104 virtual address. As a result, web client 1102 is not notified of the change and continues to use the same virtual address as previously.” Caronni I at 13:13–26.</p> <p>“In one embodiment, a virtual network supported by a physical network includes a virtual address resolution facility. The virtual address resolution facility is used to resolve a virtual IP address into a real IP address. A virtual address is registered with the virtual address resolution facility and the registration includes a real IP address, a port number and a transport protocol designation associated with a virtual address. The virtual address destination is resolved using the virtual address resolution facility and a message is sent from a virtual address in the virtual</p>

Exhibit C-29

Claims	VMware Systems
	<p>network to the real IP address indicated in the virtual address destination registration. In one aspect of the embodiment, the registration also includes an Application layer protocol designation.” Caronni II at 2:52–64.</p> <p>“In an embodiment of the present invention a virtual network is supported by a physical network and includes a virtual address resolution facility. The virtual address resolution facility is used to resolve a virtual IP address into a real IP address. A registration request is received at the virtual address resolution facility which includes a real IP address, a port address, a transport protocol designation and an Application layer protocol designation associated with a virtual address destination. The associations are stored using the virtual address resolution facility. A resolution request is received referencing a virtual address destination and the resolution request is resolved using the virtual address resolution facility and the stored associations.” Caronni II at 3:11–23.</p> <p>“The physical network 20 is also interfaced with an electronic device 30 located at the edge of a local area physical network 50 which includes devices addressable using IP addresses in the 129.63.1.0/24 range. The electronic device 30 may be a proxy server, gateway, NAT box, or other device and may perform packet filtering. Alternatively, a firewall 51 may be executed in either a hardware or software form to filter packets sent to the local area physical network 50 from the physical network 20. The firewall 51 may be located outside the electronic device 30 or alternatively may be running in software form on the electronic device 30. Those skilled in the art will recognize that a number of different types of networks may be utilized within the scope of the present invention. The networks may be the Internet, an intranet, wide area network (WAN), a local area network (LAN), a satellite network, a wireless network, or some other type of network capable of supporting a virtual network through the devices thereon.” Caronni II at 4:17–33.</p> <p>“FIG. 2 depicts a block diagram of the VARP lookup table 26 used by the virtual address resolution facility 24. The VARP lookup table 26 may be stored at any location accessible over the network. The VARP lookup table 26 includes</p>

Exhibit C-29

Claims	VMware Systems
	<p>registered virtual destination addresses located in a column 78 and corresponding associations located in a column 79. Entries 80, 82, 84, 86 and 88 include the registered virtual destination address and information associated with the registered virtual destination address. Each entry in the VARP lookup table 26 will list the registering virtual IP address 90 and associated information including a real IP address 91, a transport protocol designation 92, a port number identification 93 and an Application layer protocol designation 94. The associations are provided at the time the virtual destination address is registered in the virtual network. For example, the entry 80 for the virtual IP address 10.0.0.12 (90) includes the association of a real IP address 152.70.0.1 (91), an associated Transport layer protocol designation, UDP 92, an associated port number, 6789 (93), and an Application layer protocol designation of ‘none’ 94.” Caronni II at 5:10–29.</p> <p>“The illustrative embodiment of the present invention attempts to circumvent problems caused by packet filtering and network address translation occurring at the edge of networks through the use of higher layer protocols. FIG. 3 depicts a block diagram of the encapsulation process used to send a message from a virtual IP address to another virtual IP address located at the interior of a network at which packet filtering and/or network address translation is occurring. The virtual address resolution facility 24 is used to determine the associations registered with the destination address. The original message 111 at the Application layer 110 is formatted in a format specified in the registration of the virtual destination address if one is so indicated (e.g. HTTP). The message 111 is passed down to the Transport layer 112. The Transport layer 112 takes the original message 111 and adds a transport protocol header 113 as specified in the virtual address destination registration. The transport protocol header 113 may be a UDP header, TCP header, an X.25 header, XTP header, AppleTalk header, or other similar transport protocol header. Those skilled in the art will realize that the message may include an IPSEC header, inner IP header, UDP or TCP header and inner payload.” Caronni II at 6:32-53.</p> <p>“The illustrative embodiment of the present invention may also be utilized to send a</p>

Exhibit C-29

Claims	VMware Systems
	<p>message from a virtual address located behind a NAT box to a destination address on the same virtual network that is behind a different NAT box. Ordinarily, where the destination address is directly connected to the Internet (i.e.: in situations where the destination is not behind a NAT box), a connection, such as a TCP connection, may be established between the originator of the connection behind the NAT box and the destination. The originator's message passes the NAT box and the internal mapping of public address and port number to internal address and port number takes place. Return packets from the destination may then be received following the mapping. When the destination address is also behind a NAT box, the originator of the message is unable to directly address the destination (since the address of the destination may not be routable in the public Internet). In such a case, the present invention adds an entry to the VARP table for a third-party reflecting agent located at an address outside the NAT box. Messages that are being sent from the originating virtual address behind a NAT box to a destination which is behind a different NAT box are sent via the reflecting agent intermediary outside the NAT box and reflected to the destination.</p> <p>FIG. 5 depicts a block diagram of an environment holding a virtual network in which both the originator and destination of a message are located behind NAT boxes. An electronic device holding an originating process 182 which is part of a virtual network is interfaced with a network 180, such as the Internet, via a NAT box 186. An intended recipient for a message sent from the originating process 180 is an electronic device with a destination address 184 that is also behind a NAT box 188. A reflecting agent 200, the address of which has been added to a VARP table, acts as an intermediary to allow a connection between the originating process 182 and destination address 184.” Caronni II at 8:1–38.</p>

Exhibit C-29

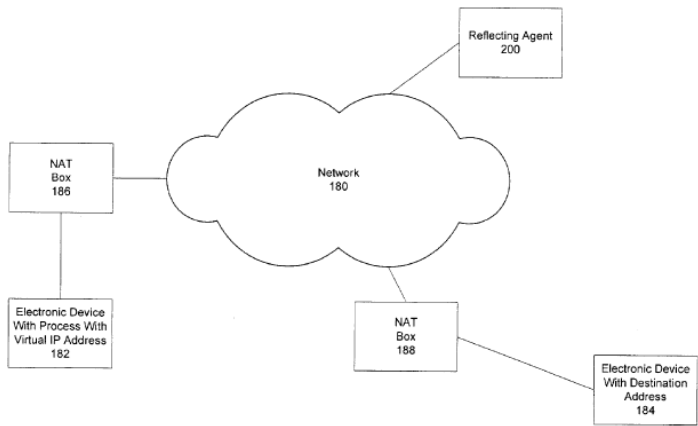
Claims	VMware Systems
	<p style="text-align: right;">Figure 5</p>  <p>Caronni II at FIG. 5.</p> <p>“Beginning at step 270, a client application requests to the VNE framework 200 to connect or send to an address of another application, such as a server application having the address 10.10.2.70: port 9001. At step 272, the VNE framework 200 requests the VNE parameters for the process corresponding to the client application from a process state storage structure. The structure that stores process state is a structure that the operating system uses to store private information about the process. Therefore, it may differ depending on the operating system. The parameters added the process state storage structure as part of the virtual network environment are listed below:</p>

Exhibit C-29

Claims	VMware Systems
	<hr/> <pre> typedef struct { ipaddr_t app_address; ipaddr_t virtual_subnet; ipaddr_t virtual_mask; ipaddr_t global_subnet; ipaddr_t global_mask; } vne_param_t; </pre> <hr/> <p>Hipp at 4:44–64.</p> <p>“Virtualization of network identity is achieved by assigning a unique virtual IP address and virtual hostname to a group of processes that make up the application instance which the instance keeps throughout its execution. This virtual network identity stays with the application instance regardless of which node the application is running on. The framework, in essence, provides a mechanism to create this virtual network identity (VNI) around the application using the virtual network parameters assigned to it. In one embodiment, the virtual network parameters include an IP address and hostname. The framework 200 ensures that the application's instance uses the virtual network parameters, transparently, so that it can be moved across machines, without modifications to the application.</p> <p>The virtual hostname resolves to the virtual IP address for both the applications registered with the VNI framework as well as those that are not registered. This may require configuration of a name service or OS host configuration files. For example, if an application instance used a virtual IP address of 10.10.0.1 and a virtual hostname of host1055, the standard hostname to IP address resolution mechanisms (e.g. DNS or the /etc/hosts file) would have to be preconfigured to resolve a query of host1055 to IP address 10.10.0.1.” Hipp at 6:1–26.</p>

Exhibit C-29

Claims	VMware Systems
	<p>To the extent that VMware Systems does not disclose any elements of this claim limitation, it would have been obvious to a person having ordinary skill in the art to combine, with the teachings of VMware Systems, those elements, as known to them or as disclosed by other prior art identified in the cover pleading to Defendant’s Invalidity Contentions.</p>
<p>35. The virtual network manager of claim 30 further comprising a join module configured to receive a join request from the agent associated with the device to indicate that the device is connected for data communication within the virtual network, the join module further configured to receive a leave request from the agent associated with the device to indicate that the device will be disconnected from data communication within the virtual network.</p>	<p>VMware Systems discloses or renders obvious the virtual network manager of claim 30 further comprising a join module configured to receive a join request from the agent associated with the device to indicate that the device is connected for data communication within the virtual network, the join module further configured to receive a leave request from the agent associated with the device to indicate that the device will be disconnected from data communication within the virtual network.</p> <p>To the extent this reference does not expressly or inherently disclose this limitation, it would have been obvious in view of the background knowledge of a person of ordinary skill in the art. This reference further renders the limitation obvious in light of other references, such as Mehta and/or Caronni I. Mehta and Caronni I concern similar subject matter as this reference, namely, virtual networks. As set forth in the accompanying contentions, joining devices with a virtual network was a well-known and commonly-used technique. Modifying this reference with Mehta’s and/or Caronni I’s teachings would have only required simple substitution, would not have required extensive experimentation, and would have yielded predictable results. A person of ordinary skill in the art would have further understood how to modify the disclosures of this reference with that of Mehta and Caronni I for example, to incorporate this limitation. <i>See, e.g.</i>,</p> <p>“Wireless systems on private networks use techniques similar to Network Address Translation (NAT) technology to send data from a wireless device to the public networking world. In a typical carrier infrastructure that uses a private network, a wireless device “registers” itself with the carrier infrastructure when it powers on (or in other circumstances, when the device attempts to initiate data services). The</p>

Exhibit C-29

Claims	VMware Systems
	<p>carrier dynamically assigns the wireless device a private non-routable address by means of DHCP (or DHCP-like) server. The information on the mapping of the transient private IP address to the device public IP address is stored in an internal carrier database and managed by carrier services such as a RADIUS server.” Mehta at [0026].</p> <p>“FIG. 4 is an example block diagram of a general purpose computer system for practicing embodiments of the Address Management Proxy System. The general purpose computer system 400 may comprise one or more server and/or client computing systems and may span distributed locations. In addition, each block may represent one or more such blocks as appropriate to a specific embodiment or may be combined with other blocks. The various blocks of the Address Management Proxy System 410 may physically reside on one or more machines, which use standard interprocess communication mechanisms to communicate with each other. In the embodiment shown, computer system 400 comprises a computer memory (“memory”) 01, a display 402, a Central Processing Unit (“CPU”) 403, and Input/Output devices 404. The Address Management Proxy System 410 is shown residing in memory 401. The components of the Address Management Proxy System 410 preferably execute on CPU 403 and manage the address mapping of wireless devices on a wireless network, as described in previous figures, to allow other wired systems to communicate with the wireless devices. Other downloaded code 405 and potentially other data repositories also reside in the memory 410, and preferably execute on one or more CPU's 403. In a typical embodiment, the AMPS 410 includes one or more DNS'/API servers 411, one or more Address Proxy/Routers 412, an Address Management Data Server 413, and Address Management Data Repositories 414. As described earlier, the AMPS may include other data repositories and components, such as a load balancer, depending upon the particular implementation.” Mehta at [0036].</p> <p>“The unique ID table 610 maps unique string names of wireless devices to the private wireless network addresses that have been assigned typically by a carrier's infrastructure. In some embodiments, the carrier infrastructure dynamically</p>

Exhibit C-29

Claims	VMware Systems
	<p>allocates, using methods similar to a DHCP protocol, a private wireless network address when the wireless device registers itself with the carrier infrastructure upon being powered on. Thus, the unique ID table 610 may be sparsely filled or entries created dynamically and then deleted dynamically as devices register and unregister with the carrier infrastructure system.” Mehta at [0053].</p> <p>“FIGS. 7A and 7B depict a flow chart of the steps performed when a node joins a Supernet. The first step performed is that the user invokes the SNlogin script and enters the Supernet name, their user ID, their password, and a requested virtual address (step 702). Of course, this information depends on the particular authentication mechanism used. Upon receiving this information, the SNlogin script performs a handshaking with SASD to authenticate this information. In this step, the user may request a particular virtual address to be used, or alternatively, the SASD may select one for them. Next, if any of the information in step 702 is not validated by SASD (step 704), processing ends. Otherwise, upon successful authentication, SASD creates an address mapping between a node ID and the real address (step 706). In this step, SASD concatenates the Supernet ID with the virtual address to create the node ID, obtains the real address of the SNlogin script by querying network services in a well-known manner, and then registers this information with the VARPd that acts as the server for this Supernet. This VARPd is identified in the configuration file.” Caronni I at 9:67–10:18.</p> <p>“FIG. 10 depicts a flow chart of the steps performed when logging a node out of a Supernet. The first step performed is for the user to run the SNlogout script and to enter a node ID (step 1002). Next, the SNlogout script requests a log out from SASD (step 1004). Upon receiving this request, SASD removes the mapping for this node from the VARPd that acts as the server for the Supernet (step 1006). SASD then informs KMS to cancel the registration of the node, and KMS terminates this KMC (step 1008). Lastly, KMS generates a new channel key for the</p>

Exhibit C-29

Claims	VMware Systems
	<p>channels on which the node was communicating (step 1010) to reduce the likelihood of an intruder being able to intercept traffic.” Caronni I at 12:61–13:5.</p> <p><i>See also</i> Caronni I at FIG. 10.</p> <p>To the extent that VMware Systems does not disclose any elements of this claim limitation, it would have been obvious to a person having ordinary skill in the art to combine, with the teachings of VMware Systems, those elements, as known to them or as disclosed by other prior art identified in the cover pleading to Defendant’s Invalidity Contentions.</p>
<p>37. The virtual network manager of claim 35 wherein the join module is further configured to maintain data to associate a virtual network address with a device in the virtual network.</p>	<p>VMware Systems discloses or renders obvious the virtual network manager of claim 35 wherein the join module is further configured to maintain data to associate a virtual network address with a device in the virtual network.</p> <p>To the extent this reference does not expressly or inherently disclose this limitation, it would have been obvious in view of the background knowledge of a person of ordinary skill in the art. This reference further renders the limitation obvious in light of other references, such as Mehta and/or Caronni I. Mehta and Caronni I concern similar subject matter as this reference, namely, virtual networks. As set forth in the accompanying contentions, joining devices with a virtual network was a well-known and commonly-used technique. Modifying this reference with Mehta’s and/or Caronni I’s teachings would have only required simple substitution, would not have required extensive experimentation, and would have yielded predictable results. A person of ordinary skill in the art would have further understood how to modify the disclosures of this reference with that of Mehta and Caronni I for example, to incorporate this limitation. <i>See, e.g.,</i></p> <p>“In one embodiment, the AMPS comprises one or more modified DNS/API servers, one or more Address Proxy/Routers, an Address Management Data Server, one or more Address Management Data Repositories, and optionally a load balancer. The</p>

Exhibit C-29

Claims	VMware Systems
	<p>AMPS DNS'/API server receives a request from a device on a public network for a particular wireless device, and returns an appropriate temporary public address, which is internally mapped to the private address of the wireless device. The public address is then usable by the device on the external public network to send data to the wireless device. The temporary public address is, for example, an address associated with one of the Address Proxy/Routers, which are connected to the external public network and have access to the private addresses on the private wireless network. In some cases, the device on the public network that desires to send data to the wireless device uses a connection-based protocol, such as TCP/IP to send data. In other cases the device uses a connection-less protocol, such as UDP (UDP/IP) to send the data.” Mehta at [0012].</p> <p>“FIG. 3 is an example block diagram of components of an example Address Management Proxy System. In one embodiment, the Address Management Proxy System (AMPS) comprises one or more modified DNS'/ API servers 302, one or more Address Proxy/Routers 305, an Address Management Data Server 303 which manages a database or other repositories such as Address Management Data Repository 304, and optionally a load balancer 301. The DNS'/API servers 302 are either individually connected to a public network 310 or are connected to the load balancer 301 which is in turn connected to public network 310. Similarly, each Address Proxy/Router 305 is also connected to the public network 310, via the routable (public) address to which data from the external network 310 is sent that is destined for the wireless devices. The DNS'/API servers 302 are either modified implementations of a DNS server to add functionality necessary to communicate with wireless devices, or are servers that implement one or more specialized APIs, as will be described further below. The DNS'/API servers 302 use the Address Management Data Server 303 to assist in mapping a unique identifier (e.g., string name) for a wireless device to a public address on public network 310. The pool of public addresses is also maintained by the Address Management Data Server 303 and Data Repository 304. The Address Management Data server 303 and Address Management Data Repository 304 may be implemented using existing database technology, for example, ODBC technology or, may be implemented as a structure</p>

Exhibit C-29

Claims	VMware Systems
	<p>such as a simple text file. One skilled in the art will recognize that any embodiment for storing a set of tables, data, lists, or mappings can be used. Each Address Proxy/Router 305 also uses the Address Management Data Server 303 or equivalent to create and update a series of routing tables that are used to assign public addresses to wireless devices as needed and to update the various mappings between public addresses and the non-routable (private) addresses of the wireless devices. The tables and mappings that are maintained on behalf of the DNS'/API servers 302 and the Address Proxy/Routers 305 by the Address Management Data Server 303 are described below with reference to FIG. 6.” Mehta at [0032].</p> <p>“One skilled in the art will recognize that the AMPS 410 may be implemented in a distributed environment that is comprised of multiple, even heterogeneous, computer systems and networks. For example, in one embodiment, the DNS'/API servers 411, the Address Proxy/Router components 412, and the Address Management Data Servers 413 with their data repositories 414 are all located in physically different computer systems. In another embodiment, various components of the AMPS 410 are hosted each on a separate server machine and may be remotely located from the tables which are stored in the address management data repository 414. In addition, under some scenarios, the entire AMPS system 410 may be hosted within a carrier's infrastructure and be completely subsumed by it. Different configurations and locations of programs and data are contemplated for use with techniques of the present invention. In example embodiments, these components may execute concurrently and asynchronously; thus the components may communicate using well-known message passing techniques. One skilled in the art will recognize that equivalent synchronous embodiments are also supported by an AMPS implementation. Also, other steps could be implemented for each routine, and in different orders, and in different routines, yet still achieve the functions of the AMPS.” Mehta at [0038].</p> <p>“FIG. 7 is an example flow diagram of an example routine provided by a DNS'/API server of the Address Management Proxy System to return a public address that corresponds to a designated unique identifier. In essence, this routine implements a</p>

Exhibit C-29

Claims	VMware Systems
	<p>DNS query or DNS-like query capability for the AMPS using a modified GetHostByName interface or a specialized API, for example GetProxyIP, as described with reference to FIG. 5. In summary, the routine dynamically allocates an appropriate Address Proxy/Router machine to associate with the wireless device and returns the public address of that machine (along with a TTL parameter and potentially other parameters). Specifically, in step 701, the routine determines the private (non-routable) address of the wireless device designated by a string parameter passed as input to the routine. For example, the string parameter may use fields such as “uniqueID.hostname.domain.tld,” which specifies a typical hierarchy of person/service on a machine named “hostname” on a domain such as a company’s network on a top level domain such as “org.” “com,” “edu,” etc. One skilled in the art will recognize that many other string parameter designations could be used. One mechanism for implementing this routine is to request information from the Address Management Data Repository. In one embodiment, the data repository stores a table that maps unique IDs to private network addresses (see Table 610 in FIG. 6). Preferably, any mechanism that is used by the AMPS stores this data in a secure manner in order to keep the wireless network addresses private. In step 702, the routine retrieves the public network address that corresponds to the private wireless network address of the designated device if one has already been assigned by the AMPS to that device and is still valid. In one embodiment, the data repository stores this mapping information between private wireless network address and public network address (see for example table 630 in FIG. 6). If a public network address has not already been assigned or is not valid, then the routine causes a new public network address to be allocated and that new public address is associated with the private wireless network address. Appropriate tables in the data repository are then updated. In step 703, the routine determines the Address Proxy/Router machine that is associated with the assigned public network address (for example using table 620 in FIG. 6). In step 704, the routine sends a request to the determined proxy/router machine to update its routing tables to map the determined public network address to the private wireless network address. In step 705, the routine updates information in the data repository to include any other</p>

Exhibit C-29

Claims	VMware Systems
	<p>connection related information (e.g., field 634 in Table 630 in FIG. 6) and indicates a Time to Live (TTL) parameter for the public-private address association (e.g., field 633 in Table 630 in FIG. 6). Once all of the tables have been updated in both the proxy/router and in the data repository, the DNS/API server returns the determined public network address of the associated Address Proxy/Router machine. As described earlier, the public address may be a (hostname, port) pair when a port-based implementation is used.” Mehta at [0057].</p> <p>“Specifically, in step 801, the routine determines (for example, from the Address Management Data Repository) the private wireless address that corresponds to the invoked public address and the TTL parameter associated with this mapping. These values can be obtained, for example, from the public-to-private address table 630 of FIG. 6. In step 802, the routine determines whether the value of the determined TTL parameter indicates that the mapping has expired, and if so, returns an error, else continues in step 803. In step 803, the routine determines the format required by the target device (the wireless network format). In step 804, the routine determines whether any protocol conversion is necessary and, if so, continues in step 805 else continues in step 806. Note that protocol conversion for a specific wireless network such as converting the data to an “HTTP” packet may be done by the proxy/router or may be done by some other component within the carrier infrastructure. One skilled in the art will recognize that these are example steps and that different formatting or different protocol conversion routines may be added or omitted as specific to the environment. In step 806, the Address Proxy/Router routine sends the data packet (which has been formatted and whose protocol is converted as necessary) to the determined associated private address of the wireless device, and returns.” Mehta at [0059].</p>

Exhibit C-29

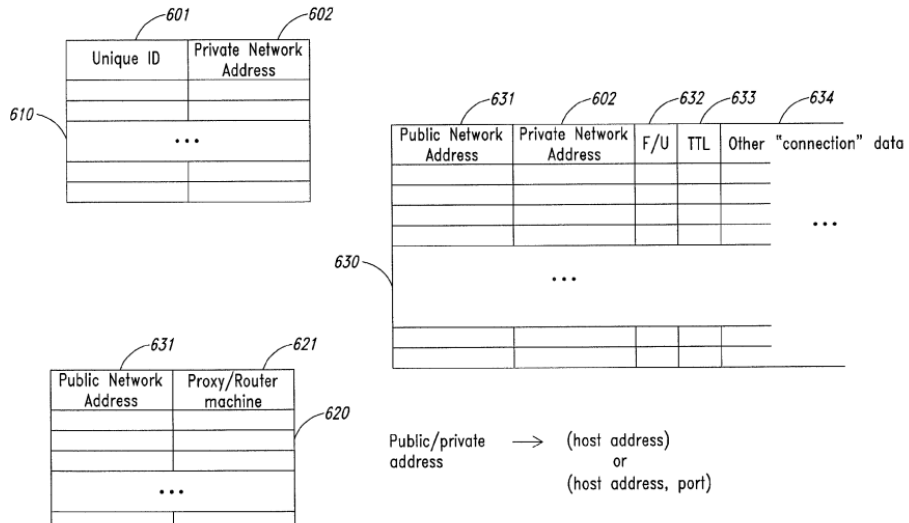
Claims	VMware Systems
	 <p style="text-align: center;"><i>Fig. 6</i></p> <p>Mehta at FIG. 6.</p> <p>“SASD 540 represents a Supernet: there is one instance of an SASD per Supernet, and it both authenticates nodes and authorizes nodes to join the Supernet. VARPD 548 has an associated component, VARPDB 551, into which it stores mappings of the internal Supernet addresses, known as a node IDs, to the network addresses recognized by the public-network infrastructure, known as the real addresses. The “node ID” may include the following: a Supernet ID (e.g., 0x123), reflecting a unique identifier of the Supernet, and a virtual address, comprising an IP address (e.g., 10.0.0.1). Although the virtual address is described in an IP address scheme, one skilled in the art will appreciate that the virtual address may be any other type</p>

Exhibit C-29

Claims	VMware Systems
	<p>addressing scheme, such as an e-mail address, IPX, or IPv6. Since the node ID includes a Supernet ID, a node will have more than one node ID when it communicates over more than one channel. The “real address” is an IP address (e.g., 10.0.0.2) that is globally unique and meaningful to the public-network infrastructure. In a Supernet, one VARPDP runs on each machine, and it may play two roles. First, a VARPDP may act as a server by storing all address mappings for a particular Supernet into its associated VARPDPB. Second, regardless of its role as a server or not, each VARPDP assists in address translation for the nodes on its machine. In this role, the VARPDP stores into its associated VARPDPB the address mappings for its nodes, and if it needs a mapping that it does not have, it will contact the VARPDP that acts as the server for the given Supernet to obtain it. The VARPDPB may also decide which virtual address to use in the translation. That is, the VARPDPB may associate a virtual address with multiple real addresses or vice versa.” Caronni I at 7:3–34.</p> <p>“FIGS. 7A and 7B depict a flow chart of the steps performed when a node joins a Supernet. The first step performed is that the user invokes the SNlogin script and enters the Supernet name, their user ID, their password, and a requested virtual address (step 702). Of course, this information depends on the particular authentication mechanism used. Upon receiving this information, the SNlogin script performs a handshaking with SASD to authenticate this information. In this step, the user may request a particular virtual address to be used, or alternatively, the SASD may select one for them. Next, if any of the information in step 702 is not validated by SASD (step 704), processing ends. Otherwise, upon successful authentication, SASD creates an address mapping between a node ID and the real address (step 706). In this step, SASD concatenates the Supernet ID with the virtual address to create the node ID, obtains the real address of the SNlogin script by querying network services in a well-known manner, and then registers this information with the VARPDP that acts as the server for this Supernet. This VARPDP is identified in the configuration file.” Caronni I at 9:67–10:18.</p>

Exhibit C-29

Claims	VMware Systems
	<p>“FIG. 10 depicts a flow chart of the steps performed when logging a node out of a Supernet. The first step performed is for the user to run the SNlogout script and to enter a node ID (step 1002). Next, the SNlogout script requests a log out from SASD (step 1004). Upon receiving this request, SASD removes the mapping for this node from the VARPD that acts as the server for the Supernet (step 1006). SASD then informs KMS to cancel the registration of the node, and KMS terminates this KMC (step 1008). Lastly, KMS generates a new channel key for the channels on which the node was communicating (step 1010) to reduce the likelihood of an intruder being able to intercept traffic.” Caronni I at 12:61–13:5.</p> <p><i>See also</i> Caronni I at FIG. 10.</p> <p>To the extent that VMware Systems does not disclose any elements of this claim limitation, it would have been obvious to a person having ordinary skill in the art to combine, with the teachings of VMware Systems, those elements, as known to them or as disclosed by other prior art identified in the cover pleading to Defendant’s Invalidity Contentions.</p>