UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CISCO SYSTEMS, INC.
Petitioner

v.

DYNAMINC MESH NETWORKS, INC.
D/B/A MESHDYNAMICS
Patent Owner

Case No. IPR2025-01304
U.S. Patent No. 7,885,243

DECLARATION OF CHRISTOPHER HANSEN, PH.D., UNDER 37 C.F.R. § 1.68 IN SUPPORT OF PETITION FOR *INTER PARTES* REVIEW

TABLE OF CONTENTS

Page

I, Christopher Hansen, Ph.D., do hereby declare as follows:

I. INTRODUCTION

1. I am making this declaration at the request of Cisco Systems, Inc. in the matter of the *Inter Partes* Review of U.S. Patent No. 7,885,243 (the "'243 Patent") to DaCosta and Dayanandan.

2. I am being compensated for my work in this matter at my standard hourly rate of $500/hour. I am also being reimbursed for reasonable and customary expenses associated with my work and testimony in this proceeding. My compensation is not contingent on the outcome of this matter or the specifics of my testimony.

3. I have been asked to provide my opinions regarding whether the subject matter of claims 1-7, 9-13 ("the Challenged Claims") of the '243 Patent would have been obvious to a person having ordinary skill in the art ("POSITA") at the time of the alleged invention, in light of the prior art. It is my opinion that the Challenged Claims would have been obvious to a POSITA.

4. In the preparation of this declaration, I have studied:

- The '243 Patent, EX1001;
- U.S. Patent Publication No. 2002/0062388 to *Ogier* et al. ("Ogier"), EX1003;

- U.S. Patent Publication No. 2002/0161917 to *Shapiro* et al. ("Shapiro"), EX1004;
- U.S. Patent Publication No. 2002/0016840 to *Herzog* et al. ("Herzog"), EX1005;
- U.S. Patent Publication No. 2003/0051051 to O'Neal ("O'Neal"), EX1006;
- U.S. Patent Publication No. 2004/0001467 to *Cromer* et al. ("Cromer"), EX1007;
- The File History of U.S. Patent No. 7,885,243, EX1008;
- The File History of U.S. Patent No. 7,420,952, EX1009;
- U.S. Patent No. 6,839,350 to *Inouchi* et al. ("Inouchi"), EX1010;
- U.S. Patent No. 6,377,782 to *Bishop* et al. ("Bishop"), EX1011;
- International Publication WO2000/035130 to *Rakoshitz* et al. ("Rakoshitz"), EX1012;
- U.S. Patent No. 6,744,775 to *Beshai* et al. ("Beshai"), EX1013;
- Infringement Contentions For '243 Patent In Dynamic Mesh Networks, Inc. d/b/a/ MeshDynamics v. Cisco Systems, Inc., No. 2:25-cv-00472, (E.D. Tex.), EX1014; and
- All other publications and materials referenced herein.

II. QUALIFICATIONS AND PROFESSIONAL EXPERIENCE

5. My complete qualifications and professional experience are described in my *Curriculum Vitae*, a copy of which is provided in Exhibit 1015 of this IPR proceeding. The following is a brief summary of my relevant qualifications and professional experience.

6. I am currently an independent technical consultant based in Los Altos, California. I work at Covariant Corporation, which is a consulting company that I own. My primary areas of expertise are in the fields of wireless networking, wireless standards development, and signal processing for wireless communications. I received a Bachelor of Science (B.S.) degree in Electrical Engineering from Rensselaer Polytechnic Institute in 1987, a Master of Science (M.S.) in Electrical Engineering from the University of Massachusetts, Amherst in 1989, and a Ph.D. in Electrical Engineering from the University of California, Los Angeles ("UCLA") in 1997.

7. I was employed by Broadcom Corporation ("Broadcom") in Sunnyvale, California from January 2000 to May 2012. From January 2000 to February 2011, I held several titles, including Sr. Staff Scientist, Engineering Manager, and Sr. Principal Scientist. My duties in these roles included, among other things, Institute of Electrical and Electronics Engineers ("IEEE") 802.11 (i.e., Wi-

Fi) physical layer ("PHY") application-specific integrated circuit ("ASIC") development. I also participated in Bluetooth standard development during this timeframe. From March 2011 to May 2012, I served as an Associate Technical Director and my duties included, among other things, performing wireless communications technology research and development ("R&D") as well as standards and intellectual property development.

8. While employed at Broadcom, I regularly attended meetings for the IEEE 802.11 Working Group. I submitted technical contributions to Task Group E ("TGe"), Task Group H ("TGh"), and Task Group N ("TGn").

9. I served on the Board of Directors of the Wireless Gigabit Alliance ("WiGig Alliance") from March 2010 to March 2012. Specifically, I served as the Board's Secretary. The WiGig Alliance was a trade association focused on the development and promotion of high-speed wireless communications technology. The WiGig Alliance collaborated for multiple years with the Wi-Fi Alliance (an industry organization that owns the "Wi-Fi" brand and certifies devices for interoperability and security), and the two organizations eventually merged in 2013.

10. I served as the Vice Chair of the IEEE 802.11ad Task Group ("TGad") from May 2011 to March 2012. TGad developed an amendment to the IEEE 802.11 wireless networking standard focused on directional multi-gigabit (i.e., DMG)

technology. In this role, I assisted the task group Chair, Eldad Perahia, and Technical Editor, Carlos Cordeiro, with the operation of the task group meetings and developing the 802.11ad draft standards amendment documents.

11. I was employed as a Senior Wireless System Architect by Apple Inc. ("Apple") in Cupertino, California from June 2012 to July 2013. During my tenure at Apple, I was responsible for, among other things, performing systems engineering for iPhone Operating System ("iOS") Wi-Fi functionality. During this time, I participated in multiple IEEE 802.11 groups, including Task Group AF ("TGaf") and the HEW Study Group, on behalf of Apple. I also regularly accessed and reviewed documents submitted by members of the IEEE 802.11 Working Group, including through the IEEE Mentor server.

12. Since August 2013, I have been employed as President of Covariant Corporation in Los Altos, California. In this role, I provide technical consulting services related to wireless communications, IEEE 802.11/Wi-Fi, signal processing technology, integrated circuits technology, and intellectual property. During the time I have been employed by Covariant, I have submitted multiple technical contributions to Task Group AY ("TGay") (responsible for 802.11ay) which led to new mmWave features, Task Group M ("TGm") (responsible for the 802.11-2020

standard), and the Integrated mmWave Study Group ("IMW SG") (the precursor for TGbq). I received awards from the IEEE for my contributions to TGay and TGm.

13. I served as a member of the Board of Directors for the Professional and Technical Consultants Association ("PATCA") in Silicon Valley, California from March 2017 to March 2024. I was President of the Board of Directors from June 2019 to March 2024. PATCA is a non-profit corporation that supports technical consultants in Silicon Valley and elsewhere.

14. Since June 2020, I have been employed as a Guest Instructor for the UCLA Department of Electrical and Computer Engineering. I teach classes on wireless communications for upper-level undergraduate students and graduate students in electrical engineering. In these classes, I use examples from the IEEE 802.11 standards.

15. I am a named inventor on over 100 issued United States patents related to wireless communications, signal processing, and integrated circuits.

16. I have been a member of the Institute of Electrical and Electronics Engineers ("IEEE") since 1983 (41 years) and am currently a Senior member. I have participated in the activities of the IEEE Standards Association ("IEEE-SA") continuously since 2000. I have participated in the IEEE 802.11 Wireless Local Area Networks ("WLAN") Working Group as a voting member since 2000.

III. LEVEL OF ORDINARY SKILL IN THE ART

17. I understand there are multiple factors relevant to determining the level of ordinary skill in the pertinent art, including (1) the levels of education and experience of persons working in the field at the time of the invention; (2) the sophistication of the technology; (3) the types of problems encountered in the field; and (4) the prior art solutions to those problems.

18. A Person of Ordinary Skill in the Art ("POSITA"), as of the claimed priority date of October 28, 2002, would have had working knowledge of the wireless mesh networking art that is pertinent to the '243 patent, including familiarity with standards such as IEEE 802.11 and mesh networking and route discovery protocols.

19. A POSITA would have had a bachelor's degree in computer science, computer engineering, electrical engineering, or equivalent training, and approximately two years of experience working in the field of network communications. Additional work experience could have been substituted for educational experience, and vice versa.

20. I met and/or exceeded these requirements for a POSITA at the time of the earliest claimed priority date of the '243 patent.

21. For purposes of this Declaration, in general, and unless otherwise noted, my statements and opinions, such as those regarding my own experience and what a POSITA would have understood or known generally (and specifically related to the references I consulted herein), reflect the knowledge that existed in the relevant field as of the claimed priority date of the ’243 patent.

IV. LEGAL STANDARD

22. I am not an attorney. In preparing and expressing my opinions and considering the subject matter of the ’243 Patent, I am relying on certain basic legal principles that Cisco’s counsel has explained to me.

23. I understand that prior art to the ’243 Patent includes patents and printed publications in the relevant art that predate the claimed priority date of the ’243 patent. For purposes of this Declaration, I am applying October 28, 2002, as the priority date of the ’243 patent.

24. I have been informed by Cisco’s counsel that a claimed invention is unpatentable under 35 U.S.C. § 103 if the differences between the claimed invention and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a POSITA. I have also been informed by Cisco’s counsel that the obviousness analysis considers factual inquiries, including

the level of ordinary skill in the art, the scope and content of the prior art, and the differences between the prior art and the claimed subject matter.

25. I have been further informed by Cisco's counsel that there are several recognized rationales for combining references or modifying a reference to show obviousness. These 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 a known technique to improve a similar device (method, or product) in the same way; (d) applying a known technique to a known device (method, or product) ready for improvement to yield predictable results; (e) choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success; and (f) some teaching, suggestion, or motivation in the prior art that would have led a POSITA to modify the prior art or to combine prior art teachings to arrive at the claimed invention.

26. I understand that one of ordinary skill in the art has ordinary creativity and is not an automaton.

27. I understand that in considering obviousness, it is important not to determine obviousness using the benefit of hindsight derived from the patent being considered.

V. OVERVIEW OF THE ’243 PATENT

28. The ’243 Patent relates to a Wireless Local Area Network (“WLAN”) designed to serve “diverse applications” with “potentially conflicting latency and throughput needs.” See EX1001, Abstract, 1:18–51. For example, the patent addresses situations where some network services—such as voice-over-IP (VoIP) or real-time video conferencing—require low latency, while others—such as file transfers or media streaming—are more tolerant of delay but require sustained high throughput. EX1001, 1:60–2:11.

29. The patent asserts that prior to the ’243 Patent, such diverse application needs were typically handled using “one access server” that “centrally managed” the allocation of network resources. EX1001, 1:40-48. EX1001, 2:22–39.

30. A centrally managed WLAN of this type places the decision-making and traffic handling for all clients and services in a single network node. This centralized approach, as described in the ’243 Patent, allegedly resulted in significant drawbacks. Specifically, the patent notes poor scalability—e.g., as the number of clients or application demands grew, the central server became a performance bottleneck. It also notes poor redundancy, because the central server represented a single point of failure: if it failed, all networked services would be disrupted. EX1001, 2:21–24 (“Unfortunately, this adversely affects scalability and

redundancy of the system: the access server is now micromanaging the network and has become a single point of failure.").

31. Finally, the approach was said to have high cost, both in terms of hardware capable of handling all processing centrally and in terms of network downtime risk. EX1001, 2:27–32 ("Another shortcoming of a centralized approach----central control and central execution-is the cost of maintaining a central control point with all intelligence and control at one location and dumb communication devices distributed in the enterprise.").

32. The '243 Patent claims to address the aforementioned conflicting application needs through a "distributed approach." EX1001, 2:33–34. According to the patent, this distributed design shifts certain network management and traffic handling functions away from a single centralized access server and into individual nodes. The '243 Patent demonstrates an exemplary network layout in Figure 1 (below), with the network including:

- "The Access server (10) [that] 'manages' the network, by setting control parameters for the network";
- "The 'Root' Node (20), [] connected to the Access Server through an Ethernet link"; and
- "AP nodes (30) that connect to the Root or other AP nodes devices to form a communications path terminating at an Ethernet link."

EX1001, 5:62–6:7.

33. Under the distributed approach, each AP node establishes a communication path in a distributed manner using what is described as a "parent selection algorithm." EX1001, 7:41–9:39.

34. In this framework, when an AP node attempts to reach the designated root node—which serves as the gateway for the mesh network—it does not rely on a centralized controller.

35. Instead, the AP node independently evaluates candidate parent AP nodes and selects one that offers a viable path toward the root. The algorithm requires the AP node to account for the performance demands of the applications running over the network. In particular, the AP node considers the "latency and throughput requirements" that the Access Server specifies for the applications associated with that AP's clients. EX1001, 6:7–11 ("To enable voice and data types of requirements to be serviced satisfactorily within the same network configuration, the access server (10) maintains a list of applications and their latency and throughput requirements."); 9:30–38 ("Combinations of both restrictions, based on the parameters set, result in networks that address both latency and throughput requirements. This was shown in FIG. 3, where the overall network configuration was somewhat mid way the two extremes. It is further evidenced in FIG. 6, where increasing the latency loss threshold results in progressively lower latency network

configurations. Nodes circled in each snap shot of the simulation show movement towards shorter routing paths as the latency cost factor is progressively increased.”).

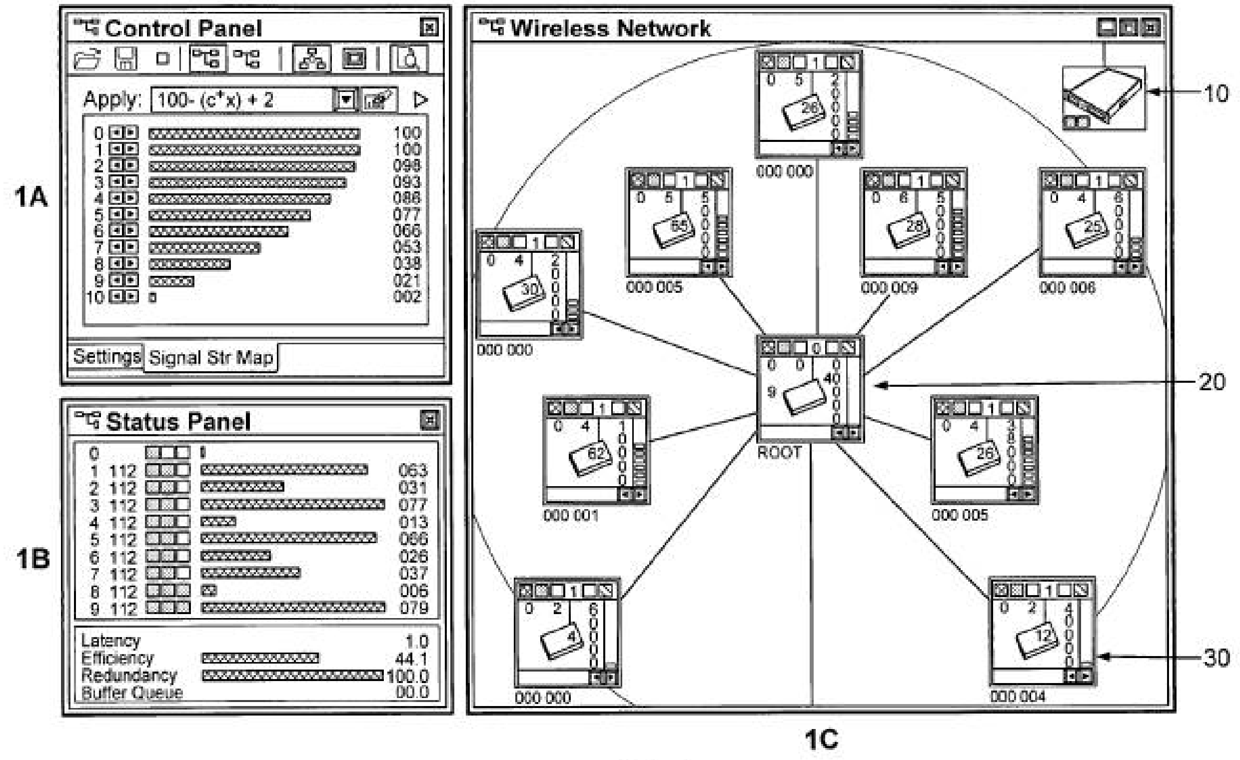


FIG. 1

EX1001, Fig. 1.

A. Prosecution History

36. The ’243 Patent was examined by the patent office as U.S. Application No. 12/154,155 (“’155 App.”), which was filed May 19, 2008, as a continuation of U.S. Application No. 10/434,948, which claims priority to U.S. Provisional Application No. 60/421,930, filed October 28, 2002. EX1001, Cover; EX1003, 65. *See also* EX1008; EX1009.

37. The ’243 Patent’s claims issued after a single round of substantive Office Action. That Office Action rejected certain claims as obvious. But it stated that claims 7–9, 11–14, and 16–17 would be allowable if rewritten in an independent form. EX1008, 150–157.

38. To overcome the rejection, Applicant cancelled the rejected claims, and amended the allowable claims in an independent form. EX1008, 163–170.

VI. CLAIM CONSTRUCTION

39. It is my understanding that in order to properly evaluate the ’243 Patent, the terms of the claims must first be interpreted.

40. It is my understanding that for the purposes of *inter partes* review, claims are to be construed under the *Phillips* standard, where claim terms are given their ordinary and customary meaning as would have been understood by a POSITA in light of the specification and prosecution history, unless the inventor has set forth a special meaning for a term.

41. I have also been informed that claim terms only need to be construed to the extent necessary to resolve the obviousness inquiry.

42. I have reviewed the entirety of the ’243 Patent, as well as its prosecution history and its parent’s prosecution history. EX1008; EX1009.

43. It is my opinion that for purposes of applying the prior art presented herein to evaluate the patentability of the Challenged Claims, no terms require formal construction aside from the term identified below.

A. Means-Plus-Function Limitation

1. "a means for switching two-way data communication from a first associated parent node to a second associated parent node based on the functioning parameters of the wireless mesh network" (All Claims)

44. I have been informed that a term claiming a "means" for performing a function is interpreted under 35 U.S.C. 112 ¶6.

45. I have identified that the claimed function of "a means for switching two-way data communication from a first associated parent node to a second associated parent node based on the functioning parameters of the wireless mesh network" is "switching two-way data communication from a first associated parent node to a second associated parent node based on the functioning parameters of the wireless mesh network."

46. The structure for this claimed function disclosed in the '243 Patent is a processor and a storage medium with instructions that perform the identified function based on latency and throughput. *See, e.g.,* EX1001, 8:1-9:12, 9:39-54, 15:15-30.

47. Applicant identified this limitation as supported by paragraph 60 of the originally submitted specification, which corresponds to 9:46-54 of the ’243 Patent. EX1001, 9:46-54. EX1008, 118.

VII. SPECIFIC GROUNDS FOR CHALLENGE

A. Ground I: Ogier In View Of Shapiro And Herzog Renders Obvious Claims 9, 12, And 13.

1. Ogier (EX1003)

48. Ogier teaches “an internetworking system” in which routing nodes “appl[y] a path selection algorithm to compute preferred paths to all possible destinations, and [] update these paths when link states are updated.” EX1003, [0038], [0198]. In other words, Ogier describes a distributed routing protocol that dynamically determines routes based on current network conditions, rather than relying on fixed or preconfigured paths. In such an approach, each node maintains knowledge of the network topology and recalculates routes when a link’s availability or cost changes.

49. Ogier shows this architecture in Figure 1, which depicts “communication sub-networks (‘subnets’) 10, 20” connected to the Internet 30. EX1003, [0038]. Within each subnet, a collection of nodes that serve different functions—including “IP hosts 12, routers 14, and a gateway 16 (collectively referred to as nodes 18).”—form the local topology. EX1003, [0038].

50. Ogier further explains that "each node 18 can establish connectivity with one or more other nodes 18 through broadcast or point-to-point links." EX1003, [0043]. This means that Ogier's network is compatible with both shared-medium links, such as a wireless broadcast channel where multiple nodes compete for access, as well as dedicated point-to-point links, such as wired Ethernet or directional wireless links.

51. For example, in Figure 1, "IP host A 12 and node B 14 have established a bi-directional link 24." EX1003, [0044]. This indicates that end-user devices as well as routers can form links within the subnet topology, and that those links support data transmission in both directions.

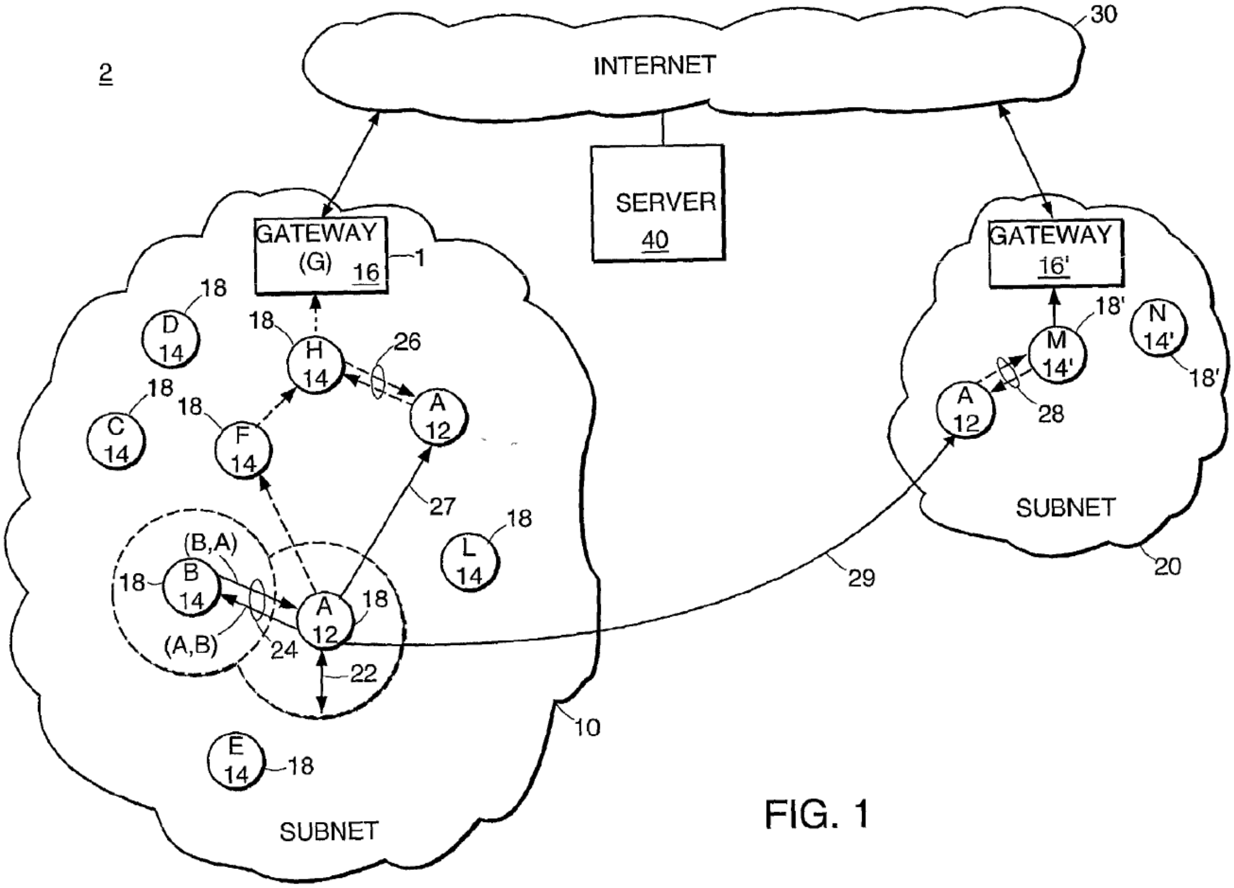


FIG. 1

EX1003, Figure 1.

52. Ogier teaches that a "routing path" for transmitting packets to a destination node can be independently selected by each node. EX1003, [0196]–[0197]. In practice, this means that no single centralized controller dictates how traffic would be forwarded across the network. Instead, each participating node applies a routing algorithm to determine the available path to any given destination, based on the node's local view of the network topology and the link-state information it maintains.

53. For example, "at least one positive cost (or metric)" is assigned to links between pairs of nodes in the subnet. EX1003, [0044]. These costs serve as quantitative measures of link quality or desirability, and they may reflect a variety of factors such as hop count, transmission delay, or any other metric relevant to routing efficiency. EX1003, [0044] ("Any technique for assigning costs to links can be used to practice the invention."). By associating a numerical cost with each link, Ogier enables the routing system to evaluate alternative paths in terms of their cumulative cost. EX1003, [0198] ("One exemplary path Selection algorithm is to apply Dijkstra's algorithm to compute shortest paths (with respect to ***cost, c***) to all destinations.").

54. Using those link costs, "[e]ach routing node 14 [] applies a path selection algorithm to compute preferred paths to all possible destinations." EX1003, [0198]. "One exemplary path selection algorithm is [the] Dijkstra's algorithm [that] compute[s] shortest paths (with respect to cost, c) to all destinations." EX1003, [0198]. In other words, every routing-capable node independently runs the path selection algorithm to identify desirable end-to-end routes—as reflected by relative costs of links—across the network. An example of such a path is illustrated below, where node F can establish a routing path to gateway

G 16 in order to route traffic from node F (or its clients) to the internet and vice versa.

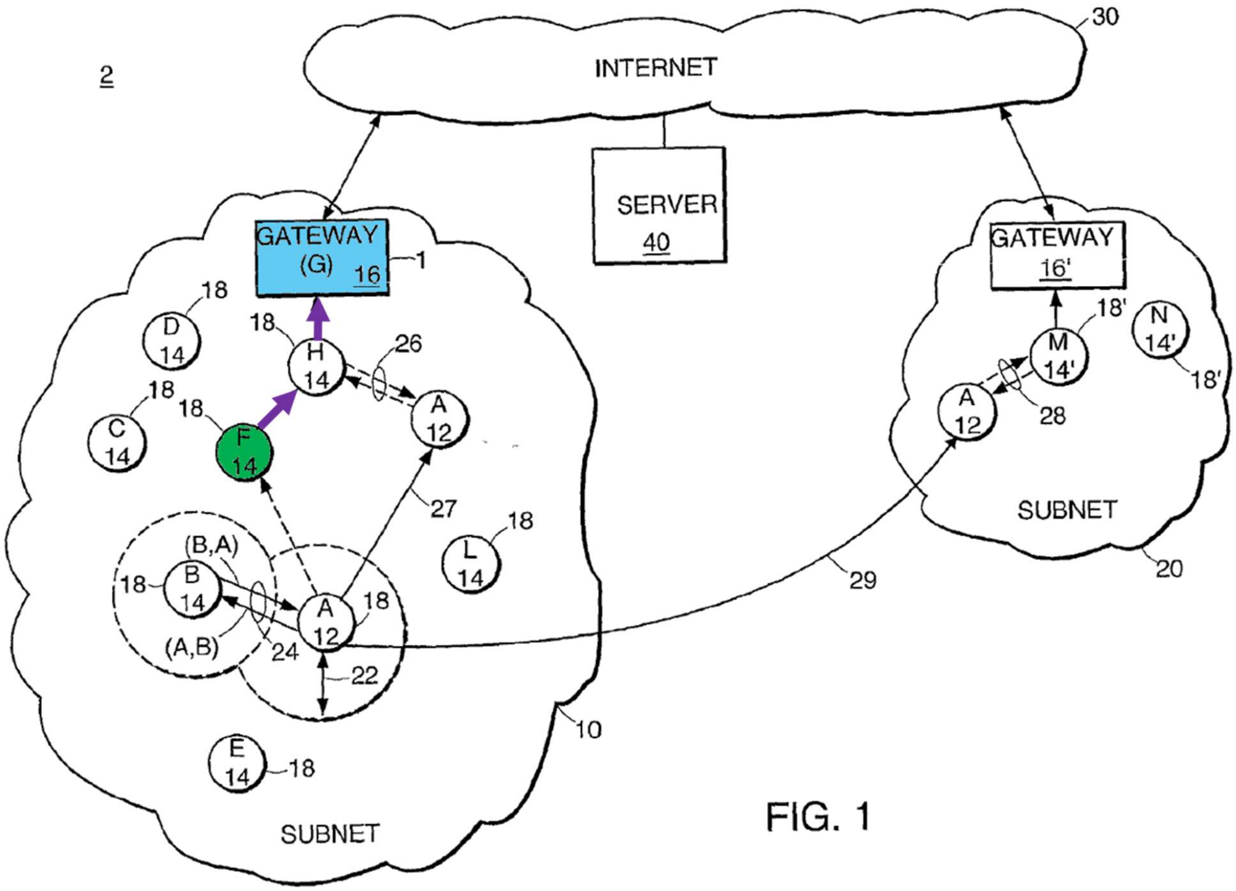


FIG. 1

EX1003, Figure 1.[1]

---

[1] In this declaration, I have added all annotations to figures unless I specifically state otherwise.

55. The determination of preferred routing paths within a subnet relies on costs that are maintained at each node in what is referred to as a “topology table.” EX1003, [0060]–[0061].

56. Ogier teaches that “each routing node 14 (or node i, when referred to generally) in the subnet 10 stores” “[a] topology table, denoted TT_i, consisting of all link-states stored at node i.” EX1003, [0060]–[0061].

57. Ogier also gives examples of how link entries are stored in said topology table; teaching that “[t]he entry for link (u, v) in this table is denoted TT_i(u, v) and includes the most recent update (u, v, c, sn) received for link (u, v),” where “[t]he component c represents the ***cost*** associated with the link.” EX1003, [0061].

58. When a routing node in Ogier’s network detects a change in the link cost of a neighboring node, it initiates a “link-state-routing protocol”—specifically, the “topology broadcast based on reverse-path forwarding (TBRPF) protocol”—to communicate the updated link-state information, such as link costs, to other routers 14. EX1003, [0047]. *See also* EX1003, [0048], [0135].

59. By initiating a link-state routing protocol at the moment of cost change, the node provides that its peers are promptly informed, enabling them to send the same information to their neighbors, and so on. In this manner, the network as a

whole updates its topology whenever a change in the network is detected by any node. EX1003, [0048] ("the TBRPF protocol performed by each of the routers 14 in the subnet 10 operates to inform a subset of the neighboring routers 14 in the subnet 10 of the current network topology and corresponding link-state information. Thus, for the examples above, each router 14 in the subnet 10 that detects a change in a link to node A 12, (e.g., node B 14 in the cost of the link (B, A)), operates as the source (i.e., source node) of an update. Each source node sends a message to a neighbor of that source node, informing the neighbor of the update to that link. Each router 14 receiving the update may subsequently forward the update to zero or more neighbor nodes, until the change in the topology of the subnet 10 disseminates to the appropriate routers 14 in the subnet 10.").

60. Ogier discloses a "source node," which is a node executing the TBRPF protocol. The source node can distribute its link costs to other nodes in the subnet along a tree structured graph called a "path tree." EX1003, Abstract, [0010]. "Each path tree has the source node as a root node, a parent node, and zero or more children nodes." EX1003, Abstract.

61. The path tree can take on different structures, such as a "minimum-hop-path tree," a "shortest path tree," or "other types of trees," and that the tree is

"updated dynamically using the topology and link-state information that are received along the []path trees themselves." EX1003, [0058], [0064], [0084].

62. The choice of tree structure influences what the nodes prioritize when making routing decisions: a "minimum-hop-path tree" selects for hop count, while a "shortest path tree" minimizes aggregate link cost, which may be configured to capture factors such as delay, or throughput, as explained further in Section VII.A.4. EX1003, [0044], [0058], [0084].

63. The path tree is updated by "running a shortest-path algorithm such as Dijkstra's algorithm." EX1003, [0084]. Because the "topology and link-state information" is maintained in the topology table TT_i (EX1003, [0061]), the "shortest path tree" embodiment of TBRPF (EX1003, [0058]) would use "a shortest-path algorithm such as Dijkstra's algorithm" (EX1003, [0084]) by taking information contained in topology table TT_i as inputs to compute the "shortest path tree." In this manner, TT_i serves as the local repository of link states from which the node can construct and update its routing decisions.

64. Just as any other routing node in the network, gateway 16 can serve as the "source node" of the path tree. Ogier teaches that "each router 14 in the subnet 10 runs a link-state-routing protocol" (e.g., TBRPF), and "[t]he gateway 16 is a

particular type of routing node 14 that connects the subnet 10 to the Internet 30." EX1003, [0040], [0047].

65. Accordingly, in addition to connecting the subnet to the broader internet, the gateway node participates in the topology discovery and dissemination process. As a routing node executing TBRPF, the gateway would also act as a "source node" when it is responsible for communicating a link state update that it detected locally.

66. When the gateway node serves as the source node of a path tree, the nodes of the subnet would be arranged path tree (shown by the dotted red line) as depicted below, with gateway G 16 as the root node, and the rest of the nodes 14 (e.g., **F**, **H**) arranged in a parent-child relationship.

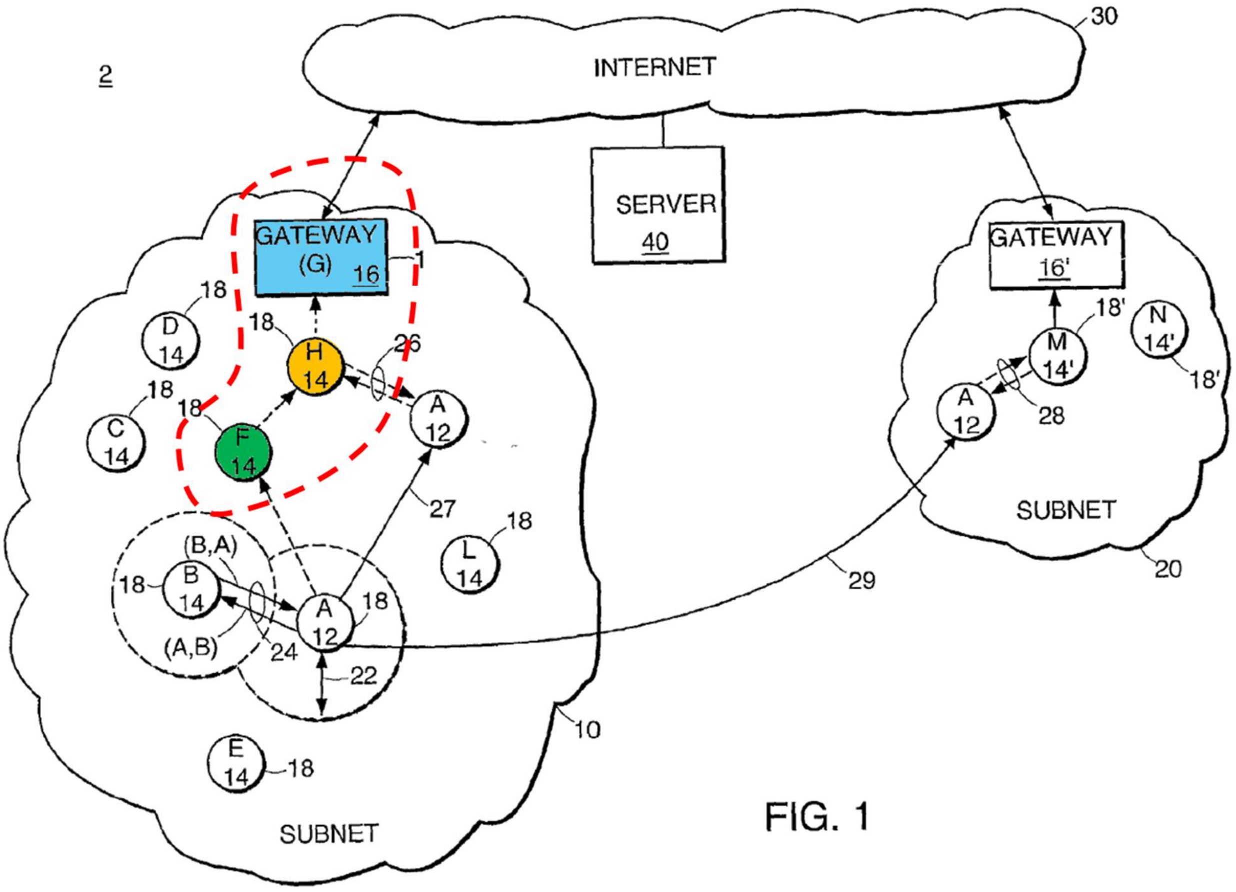


FIG. 1

EX1003, Figure 1.

2. Shapiro (EX1004)

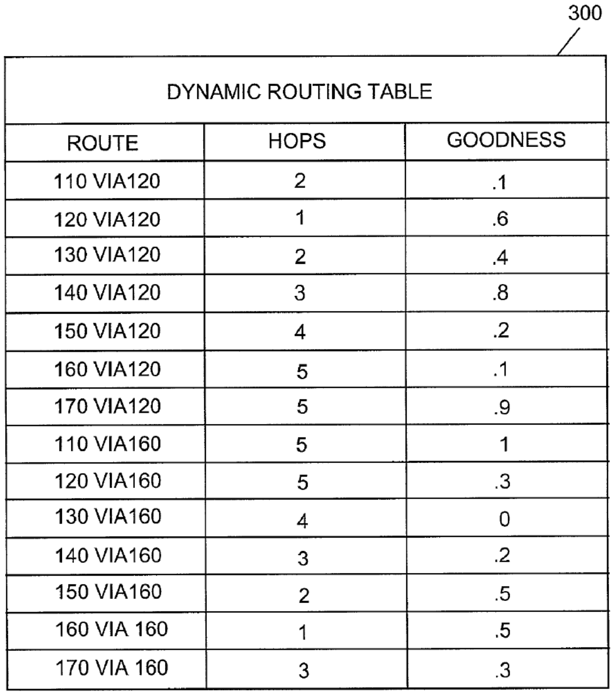
67. Shapiro discloses “systems for dynamically routing data through a network of nodes” that “takes into account the quality or speed of a link.” EX1004, [0001], [0008]. Rather than relying solely on static hop-count metrics or predetermined routes, Shapiro’s system introduces adaptive routing decisions that respond to real-time link performance characteristics. For example, Shapiro’s nodes can account factors such as throughput, giving the system has more flexibility to

select paths more suitable for particular network conditions. EX1004, [0048]. ("In the exemplary embodiment of the invention, goodness factor is based on a decaying average of periodically sampled throughput for a node.").

68. In Shapiro's system, each node maintains a "dynamic routing table" with "routing information" (EX1004, [0009], [0046]), which it uses to "determine the best route to send the data" to the destination node (EX1004, [0009], [0049]). A POSITA would have understood that this architecture enables distributed decision-making, as each node independently computes its forwarding choices based on the most up-to-date local information.

69. To determine the "best route", a node calculates "an efficiency factor" "for each possible route" to the destination, in order to select the route with the best efficiency factor. EX1004, [0053]. *See also* EX1004, [0046]–[0048]. This approach reflects a multi-factor framework in which routing decisions are not restricted to a single criterion. Instead, the "efficiency factor" acts as a composite metric that can incorporate multiple dimensions of network performance, which may be computed using various types of information. EX1004, [0053]. This calculation would have also been easily modified to incorporate other criteria, as explained further in SectionVII.A.4.a).

70. As shown in Figure 3, for each candidate route, the dynamic routing table stores: (1) the "number of nodes that must be traversed for a data packet to travel from the current node to the destination node" (referred to as "Hops"), and (2) a "goodness factor" that "represents any number of qualitative and quantitative feature of the corresponding route." EX1004, [0009], [0047], [0048].



DYNAMIC ROUTING TABLE

ROUTE	HOPS	GOODNESS
110 VIA120	2	.1
120 VIA120	1	.6
130 VIA120	2	.4
140 VIA120	3	.8
150 VIA120	4	.2
160 VIA120	5	.1
170 VIA120	5	.9
110 VIA160	5	1
120 VIA160	5	.3
130 VIA160	4	0
140 VIA160	3	.2
150 VIA160	2	.5
160 VIA 160	1	.5
170 VIA 160	3	.3

EX1004, Figure 3.

71. The goodness factor represents "characteristic[s] of a route," and is flexible, as it "may be a function of any number of factors." EX1004, [0048]. This flexibility allows the routing network to adapt to diverse conditions and policy objectives. For example, the goodness factor may be computed based on "a decaying average of periodically sampled ***throughput*** for a node." EX1004, [0048].

72. Shapiro also teaches that the "[g]oodness factor may be used by a network manager to encourage traffic via a certain route or away from a certain route." EX1004, [0048]. This functionality would have allowed network administrators to introduce policy-driven parameters into the routing decisions made by nodes.

73. Finally, Shapiro teaches that in some circumstances, the goodness factor reflects the characteristics of all nodes and links along a particular route. EX1004, [0045] ("The goodness factor G_{170} is representative of the quality of the link from node 170 to node 150."). A POSITA would have understood that such a factor would also be configured to represent single-link connections, and therefore would represent a characteristic of a particular link. EX1004, [0045] ("The goodness factor G_{150} is representative of the quality of the link from node 150 to node 140.").

3. Herzog (EX1005)

74. Herzog describes a "Policy Based Networking (PBN)" technique that "control[s] network operation and influenc[es] the way packets are handled by network nodes based on high layer criteria." EX1005, [0005]. Herzog explains:

> In general, with PBN, network administrators first define networking goals (i.e., "network policy"). Those networking goals are then provided to a policy system which automates and translates the policy

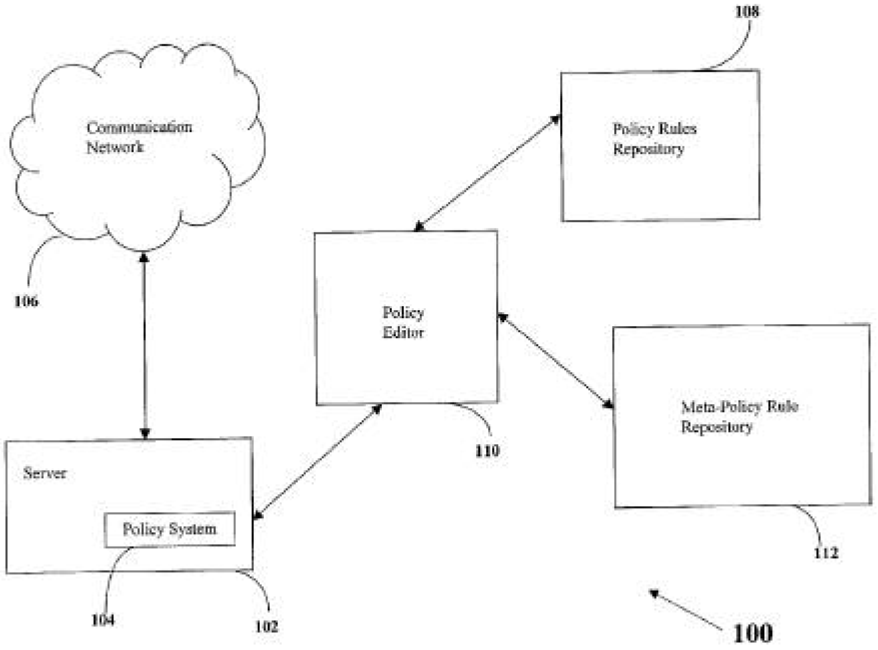
> into a set of lower-level instructions. Network devices understand the instructions, and the specified goals thus can be accomplished.

EX1005, [0005]. *See also* EX1005, [0023].

75. The PBN introduces the ability for routing and forwarding decisions to be shaped by organizational objectives, application requirements, or user-defined constraints. Such network policies were known to enforce performance specifications. For example, Herzog teaches one policy that requires that data packets are not subject to "a delay no greater than a certain amount, a bandwidth no less than a certain amount, and/or priority higher than some or all other packets." EX1005, [0005].

76. A POSITA would have recognized that these types of performance specifications as a form of quality-of-service (QoS) requirements, which provide that data-dense or latency-sensitive traffic—such as real-time voice, video, or control signaling—receives preferential treatment across the network. EX1005, [0003] ("nodes may allow or deny the packets access to network resources, provide preferential treatment of the packets, or provide a lower quality of service, for example.").

77. Herzog also explains that the "policy system"—the system responsible for translating policy objectives into instructions that individual nodes understand—typically resides on a server, as illustrated in Figure 1 (below). EX1005, [0029].



EX1005, Figure 1.

4. Ogier-Shapiro-Herzog Combination

a) Computing Link Costs Based On An Additional Goodness Factor, As Taught By Shapiro

78. As discussed in Section VII.A.1 [Ogier], Ogier teaches that the nodes in its network each select a routing path by "apply[ing] Dijkstra's algorithm to compute shortest path[] (with respect to cost, c) to [the destination]" (EX1003, [0198]), and that the TBRPF path tree may also be a "shortest path tree" (EX1003, [0058]) determined by "run[ning] a shortest-path algorithm such as Dijkstra's algorithm" (EX1003, [0084]) on the topology table TT_i.

79. This teaching underscores that Ogier's protocol is not limited to a single cost function but is designed to operate on a cost parameter "c," the definition of which is left to the system designer.

80. In Ogier, the cost of a link between two nodes—used to compute both the shortest-path tree and shortest routing paths —"can be ***one***, for minimum-hop routing, or the ***link delay plus a constant bias***," but may also be set using "[a]ny technique." EX1003, [0044]. A POSITA would have recognized this open-ended design choice as an invitation to incorporate more application-specific cost metrics into the routing process. In practice, such flexibility is desirable when a single fixed metric is insufficient to capture real-world performance differences between links.

81. Incorporating additional link metrics into such "link cost" variables was well known. Shapiro itself indicates this; as explained in Section VII.A.2 [Shapiro], Shapiro teaches selecting the "best route" based on an "efficiency factor," which depends on a "goodness factor" representing "any number of qualitative and quantitative feature[s] of the corresponding route." EX1004, [0048], [0053]–[0054]. The goodness factor may capture, for example, "a decaying average of periodically sampled ***throughput*** for a node." EX1004, [0048]. A POSITA would have understood that Shapiro's "goodness factor" is the type of flexible metric that Ogier

contemplated when it taught that cost may be computed using "[a]ny technique." EX1003, [0044].

82. It would have been obvious to combine Ogier with Shapiro by incorporating Shapiro's "goodness factor" into Ogier's link cost. EX1004, [0048]. Since Shapiro's efficiency factor is computed as the weighted sum of the goodness factor and another feature, it would have been obvious to modify Ogier's link cost in a commensurate manner; i.e., to calculate link cost as a weighted sum of: (1) "one, for minimum-hop routing, or the link delay plus a constant bias" as taught by Ogier (EX1003, [0044]); and (2) the goodness factor as taught by Shapiro (EX1004, [0048]). *See* EX1004, [0053] ("In the exemplary embodiment of the invention, the ***algorithm*** yields the result of the calculation constant, K, multiplied by the Hops, plus the goodness factor, or, ***Efficiency=K(Hops)+goodness factor***.").

83. This modification would allow the routing protocol to retain Ogier's structure of using a cost basis (hop count or link delay) while enhancing it with Shapiro's context-sensitive goodness metric, thereby improving responsiveness to varying link conditions. The two link cost algorithms would therefore be:

- $LC_1 = 1 + W*GF$; or
- $LC_2 = (LD + bias) + W*GF$

where LC represents link cost, LD represents link delay, GF represents the goodness factor, and W represents the relative weight between Ogier's cost (i.e., 1 or LD + bias) and the goodness factor. EX1004, [0053]. LC_1 is selected when Ogier's minimum-hop routing teaching is preferred; and LC_2 is selected when Ogier's link delay (latency) teaching is preferred.

84. Shapiro provides a few examples of goodness factors that capture characteristics of an entire route (e.g., all nodes and links along the route), as opposed to an individual link. But a POSITA would have understood that those same characteristics also apply to an individual link, since a link is a one-hop route.

85. When Ogier's path tree generation algorithm and "path selection algorithm" use LC_1 as the link cost, the resulting path tree or routing path would reflect both Ogier's "minimum-hop routing" and the characteristics captured by the goodness factor (such as periodically sampled throughput). When LC_2 is used, the resulting tree or path reflects both link delay (i.e., latency) and the goodness factor. EX1003, [0198]. A POSITA would have further recognized that the "network manager" would be used to increase or decrease the relative weight of the goodness factor "to encourage traffic via a certain route or away from a certain route." EX1004, [0048].

a. Reasons To Combine Ogier And Shapiro

86. A POSITA would have been motivated to incorporate Shapiro's node throughput into Ogier's link cost.

87. ***First***, Ogier provides motivation for such a combination. A POSITA would have recognized that Ogier leaves open the definition of link cost by teaching that "***[a]ny technique*** for assigning costs to links can be used to practice the invention." EX1003, [0044]. This statement signals that the protocol may accommodate a wide range of cost metrics beyond solely hop count or delay, and thus invites the incorporation of additional performance-related metrics into the link costs calculations.

88. A POSITA would have readily recognized that Shapiro's goodness factor provides such a feature, as it enables route selection based on dynamic characteristics such as aggregate node throughput along a routing path. EX1004, [0048]. ("In the exemplary embodiment of the invention, goodness factor is based on a decaying average of periodically sampled throughput for a node.").

89. Additionally, Ogier acknowledges throughput is a significant factor affecting communication performance: "[a] factor in determining the packet length is the tradeoff between data ***throughput*** and the percentage of packet loss." EX1003, [0352]. Therefore, Shapiro's teaching of a throughput-based goodness factor would

have been viewed as complementary with Ogier's recognition that throughput is a critical performance variable. Incorporating this factor into Ogier's cost metric would have been a natural and straightforward way to achieve more context-sensitive routing, in line with both references' teachings.

90. ***Second***, Shapiro also provides motivation for the combination. Shapiro notes "a need for dynamic routing of data that takes into account the quality or speed of a link." EX1004, [0008]. To solve this, Shapiro teaches that routes ought to be selected based on a "goodness factor." EX1004, [0048]. That goodness factor is a flexible and extensible metric that "represents any number of qualitative and quantitative feature of the corresponding route," including, for example, "a decaying average of periodically sampled throughput for a node." EX1004, [0048]. Shapiro further explains that this mechanism allows a network manager to "encourage traffic via a certain route or away from a certain route." EX1004, [0048]. A POSITA would therefore have been motivated to incorporate Shapiro's goodness factor into Ogier's link cost, as the modification would enable Ogier's system to create routing paths and path trees that "tak[e] into account the quality or speed of a link," as Shapiro teaches. EX1004, [0008].

91. ***Third***, Ogier and Shapiro are compatible and readily combinable. Both references are directed to solving the same problem in wireless communication

networks: selecting routing paths or path trees that optimize a link-cost-based metric. Ogier teaches that shortest paths may be determined by applying Dijkstra's algorithm "with respect to cost, c." *See* EX1003, [0198] ("One exemplary path selection algorithm is to apply Dijkstra's algorithm to compute shortest paths (with respect to ***cost, c***) to all destinations."); [0084]. Dijkstra's algorithm was a well-known optimization algorithm that was well-suited for computing applications.

92. Similarly, Shapiro teaches selecting a "best route" based on an "efficiency factor." EX1004, [0053] ("[T]he dynamic routing table 300 is used to lookup the best route …. An algorithm calculates, for each possible route in the table, an ***efficiency factor***."). A POSITA would have readily understood that Shapiro's efficiency factor calculation would be used to calculate Ogier's cost metric "c".

93. Both Ogier and Shapiro teach that their respective cost metrics reflect the status or quality of links between nodes. For example, Ogier describes "assigning costs to links" and explains that when a node changes its physical location in a subnet, "[s]uch movement by one node 18 does not necessarily result in breaking a link, but may diminish the ***quality of the communications*** with another node 18 ***over that link***. In this case, a ***cost of that link has increased***." EX1003, [0044], [0045]. Ogier further states that "[a] ***cost*** of infinity represents a failed link." EX1003, [0108].

94. Shapiro likewise discloses that its cost metric captures link quality, teaching that "[t]he goodness factor G170 is representative of ***the quality of the link*** from node 170 to node 150," (EX1004, [0045]) and that "[t]he goodness factor represents any number of qualitative and quantitative feature of the corresponding route." EX1004, [0048]. These disclosures are consistent with the general understanding in the art, as it was well known that "[t]he cost of a link may be defined according to several criteria including such qualities as reliability and delay." EX1013, 1:27–29. A POSITA would therefore have recognized that both Ogier and Shapiro operate using link-quality–sensitive cost metrics, further confirming their compatibility.

95. For the aforementioned reasons, Shapiro's goodness factor, which reflects quality of service characteristics such as throughput, would have been obviously applicable to Ogier's link cost calculation. Employing Shapiro's goodness factor to calculate Ogier's link costs in the aforementioned equations would have merely amounted to combining one prior art element (Shapiro's goodness factor) with another prior art element (Ogier's cost, 1 or "link delay plus a constant bias") according to known methods (calculating a weighted sum of the variables, as described by Shapiro) to yield the predictable results of assigning the link cost for Ogier's route path selection based on the combined elements.

96. A POSITA would have reasonably expected to succeed in incorporating Shapiro's goodness factor into Ogier's link cost as described for three reasons.

97. ***First***, the combination of Ogier and Shapiro amounts to nothing more than a routine modification—specifically of the link costs that are used as inputs in Ogier's route-selection and path-tree generation processes—which would have predictably resulted in the computation of paths that optimized for those modified costs. Ogier explains that routing paths are selected using Dijkstra's algorithm. EX1003, [0198] ("apply ***Dijkstra's algorithm*** to compute shortest paths (with respect to cost, c) to all destinations."); [0084] ("Node i then computes (step 104) the parent nodes p i(u) for all potential source nodes src by running a shortest-path algorithm such as Dijkstra's algorithm.").

98. A POSITA would have recognized that incorporating Shapiro's link-quality metric into Ogier's cost function would not have altered the operation of Dijkstra's algorithm itself. Rather, it was well understood that Dijkstra's algorithm produces optimal paths with respect to whatever cost metric is used as an input for calculation. EX1013, 13:51–67 ("Typically, an algorithm is used to generate the overall routing table of FIG. 5. One such algorithm is the Dijkstra algorithm…. Using the algorithm, finding the shortest (i.e.,the route which minimi[z]es a metric)

route for travelling from a given vertex on a graph to every other vertex is possible.… In the case where the metric represents the cost of each link, the algorithm returns an indication of the least cost route from the root vertex to the particular vertex as well as the overall cost of the route.”). Thus, applying Dijkstra’s algorithm in the combined Ogier-Shapiro system would have been a routine and predictable exercise, yielding shortest paths that reflect both Ogier’s structural cost (e.g., hop count or delay) and Shapiro’s goodness factor (e.g., throughput).

99. ***Second***, both Ogier and Shapiro teach route selection methods contemplate flexibility in how routing metrics are defined. Ogier states that “***[a]ny technique*** for assigning costs to links can be used to practice the invention.” EX1003, [0044]. Therefore, Ogier’s path-selection process can accommodate different cost metrics beyond hop count or delay, so long as they can be expressed numerically and applied within Dijkstra’s algorithm.

100. Similarly, Shapiro explains that its goodness factor—the metric upon which its route selection is based—“represents any number of qualitative and quantitative feature of the corresponding route.” EX1004, [0048]. Taken together, a POSITA would have understood from these teachings that modifying Ogier’s link cost to incorporate Shapiro’s goodness factor would operate as intended. A POSITA would therefore have had a reasonable expectation that the link cost in the Ogier-

Shapiro system would be effective in selecting route paths that would account for multiple metrics, consistent with the teachings of both references.

101. ***Third***, a POSITA would have recognized that Ogier's path selection process is implemented by programming the routers in Ogier's network, as Ogier relies on the use of Dijkstra's algorithm. The Ogier–Shapiro combination would therefore have required nothing more than writing software instructions to adjust how link costs are computed prior to running Dijkstra's algorithm. Such a modification was squarely within the capacity of a POSITA at the time.

b) Setting Cost Criteria Using A Policy Server, As Taught By Herzog

102. As discussed above, Shapiro teaches a "network manager" that can use the "goodness factor" to "encourage traffic via a certain route or away from a certain route." EX1004, [0048]. A POSITA would have therefore understood that, in the context of the Ogier–Shapiro combination, the network manager would be responsible for defining the parameters used in the link-cost calculation—such as whether LC_1 or LC_2 should be calculated, how the goodness factor is defined, and the appropriate value of the weight W.

103. However, while both Ogier and Shapiro describe the role of these parameters in guiding path selection, neither reference discloses a mechanism by

which the "network manager" would actually provide such parameters to Ogier's nodes.

104. Using a network manager to provide parameters to a network of nodes was well known in the art. For example, Herzog discloses "network administrators" that define a "network policy" and provide it to a policy system residing on a server. EX1005, [0005]. A POSITA would have understood such goals may include requirements concerning delay, bandwidth, or priority of traffic.

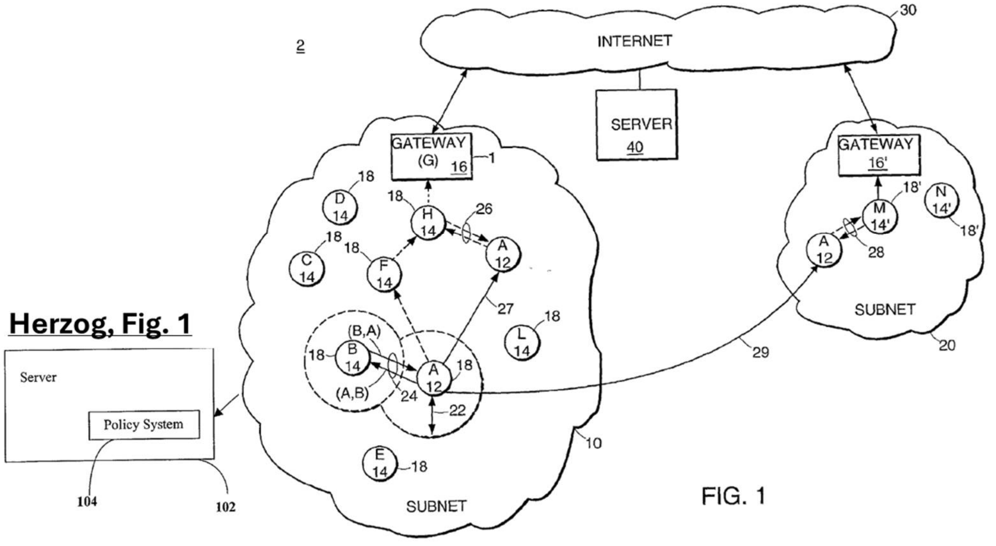
105. The policy server then "translates the policy into a set of lower-level instructions," which are distributed to the network nodes for implementation. EX1005, [0005], [0029] ("The policy system 104 … translates the policy rules into a set of lower-level instructions that network devices understand.").

106. A POSITA would have understood that Herzog's disclosure elaborates on the "network manager" identified by Ogier and Shapiro by providing a server-based method through which the network's parameters—such as the choice between LC_1 and LC_2, the definition of the goodness factor, and the value of W—would be supplied to and implemented by the nodes.

107. Setting such parameters from a single server would have been convenient, given that a manager would set the high level policies from a single point, with the subnet reconfiguring itself to comply with the policies.

108. Accordingly, it would have been obvious to combine Ogier and Shapiro with Herzog in order to incorporate the policy server. Herzog's policy server would supply the nodes of the combined system with "a set of lower-level instructions" specifying key parameters for link-cost computation, including whether LC_1 or LC_2 should be used, the definition of the goodness factor, and the value of the weight *W*. By integrating Herzog's policy system with Ogier's path-tree routing and Shapiro's goodness-factor metric, the combined system (the "Ogier–Shapiro–Herzog combination") enables routing decisions that are based on both network performance characteristics and administrator-defined policies. The structure of this combination is illustrated below.

Ogier-Shapiro-Herzog Combination



EX1003, Figure 1; EX1005, Figure 1.

a. Reasons To Combine Ogier And Shapiro With Herzog

109. A POSITA would have been motivated to implement Herzog's policy server to distribute lower level instructions to nodes in the Ogier-Shapiro combination.

110. ***First***, although neither Ogier nor Shapiro teach how a "network manager" assigns parameters to routing nodes for link costs computation, Herzog teaches a policy server capable of performing this function. This would have motivated a POSITA to use Herzog's policy server to provide "a set of lower-level instructions" to Ogier's nodes. EX1005, [0005]. The nodes, in turn would "understand the instructions" and execute them. EX1005, [0005].

111. Herzog's teaching makes clear that its policy server is appropriate for this context, because it is designed to manage network policies relating to performance characteristics such as delay and bandwidth. EX1005, [0005] ("The action or goal could be to guarantee those packets some preferential treatment such as a delay no greater than a certain amount, a bandwidth no less than a certain amount, and/or priority higher than some or all other packets.").

112. Thus, Herzog provides a mechanism needed to translate high-level network objectives into the parameters required for Ogier's cost-based routing process enhanced with Shapiro's goodness factor.

113. ***Second***, combining the Ogier–Shapiro system with Herzog's policy server would have merely involved the integration of known prior art elements—namely, the Herzog's policy server and Ogier–Shapiro system—using methods known in the art at the time. In practice, this would simply require establishing a communication link between Herzog's policy server and the Ogier–Shapiro subnet. The predictable result of such a combination is that the policy server would distribute link-cost parameters, thereby enabling Ogier's nodes to compute link costs in accordance with those parameters. A POSITA would have regarded this as a straightforward application of established networking techniques.

114. ***Third***, both Ogier and Shapiro teach networks that are subject to administrative control. Ogier discloses that its system includes "an administrative authority for the subnet 10" responsible for enforcing network policies. EX1003, [0302] ("… if an ***administrative authority*** for the subnet 10 wishes to enforce a ***policy***…"), [0315] ("The ***administrative authority*** … may institute a ***policy*** …"), [0325] ("…a network ***administrative authority*** wishes to configure…."). Likewise, Shapiro describes the presence of a "network manager" that directs and influences network traffic. EX1004, [0048].

115. In view of these teachings, a POSITA would have been motivated to incorporate a policy server into the Ogier–Shapiro combination to enforce network

policies—such as policies governing link-cost computation—and would have readily understood that Herzog's policy server was applicable for distributing parameters related to network policies.

116. A POSITA would have reasonably expected success in incorporating Herzog's policy server into the Ogier-Shapiro's system as described above for three reasons.

117. ***First***, Herzog's policy server would have operated in a predictable manner within the Ogier–Shapiro system. For example, if a network manager wished influence traffic flow—as taught by Shapiro—by adjusting the goodness factor to "encourage traffic via a certain route" (EX1004, [0048]) characterized by higher throughput and lower delay, the manager would define such a "network policy" and provide it to Herzog's policy server. Herzog's policy server then operationalizes that goal by then "translat[ing] the policy into a set of lower-level instructions" (e.g. parameters instructing nodes to compute link cost using LC_2 with the goodness factor defined as node throughput (EX1004, [0048])) and delivering those instructions to Ogier's nodes. EX1005, [0005].

118. Conversely, if the network manager instead defined a policy favoring paths with fewer hops, the server would then translate that policy into parameters that instruct the nodes to compute link cost using LC_1 with the weight W set at zero,

such that the goodness factor would not affect the cost. In both scenarios, Ogier's nodes would "understand the instructions" and predictably compute paths consistent with the policy goal by applying Dijkstra's algorithm to the adjusted link costs.

119. Therefore, Herzog's policy server yields only the expected results of allowing a network manager to define a policy, automatically translating that policy into parameters, and then having those parameters understood and implemented by Ogier's nodes. A POSITA would have regarded this outcome as predictable, given that each reference already disclosed its respective functionality: Ogier's path-tree computation, Shapiro's goodness factor for link-quality sensitivity, and Herzog's policy server for translating and distributing administrator-defined policies.

120. ***Second***, as explained above, both Ogier and Shapiro discuss that an administrator manages the nodes in their networks. Accordingly, it would have been apparent that Herzog's policy server was compatible with the Ogier–Shapiro system, and would predictably function to deliver routing-related parameters to Ogier's nodes. Incorporating Herzog's policy server into the Ogier–Shapiro system would not have required any significant architectural modification. Rather, it would have simply involved connecting Herzog's policy server to the Ogier–Shapiro subnet through a conventional communications link, after which the policy server would

translate administrator-defined policies into parameters and transmit them to the nodes for implementation.

121. ***Third***, a POSITA would have understood that Herzog's policy server was designed to bridge the gap between administrator-defined policies and router-level implementation by producing instructions that network nodes can "understand." EX1005, [0005]. Accordingly, a POSITA would have reasonably expected that the same infrastructure would be applied to the Ogier–Shapiro system to communicate link-cost parameters to nodes executing Ogier's shortest-path computations. Moreover, since Ogier, Shapiro, and Herzog all address the same challenges in the same technical domain—data packet routing and traffic management in communications networks—a POSITA would have had a reasonable expectation of success in integrating the three systems into a unified whole. Each reference contributes a well-understood component: Ogier's path-tree routing and optimization algorithm, Shapiro's link-quality–sensitive goodness factor, and Herzog's policy server. A POSITA would have viewed their integration as straightforward and predictable.

5. Claim 9

a) [9Pre] A wireless mesh network comprising:

122. If the preamble is limiting, Ogier-Shapiro-Herzog teaches [9Pre].

123. ***First***, Ogier discloses a "*mesh network*." Ogier teaches a network topology in which nodes can connect to multiple other nodes, creating a multi-hop structure. For example, Ogier describes a "method for disseminating topology and link-state information over a ***multi-hop network*** comprised of nodes." EX1003, [0010]. Ogier explains that "***[e]ach subnet 10***, 20 includes one or more networks." EX1003, [0040]. In particular, "[t]he subnet 10 includes IP hosts 12, routers 14, and a gateway 16 (collectively referred to as nodes 18)," (EX1003, [0040]), and "each node 18 can establish connectivity with one or more other nodes 18." EX1003, [0043]. Taken together, these disclosures make clear that the subnet 10 is a "*mesh network*" of nodes 18, since each node is capable of forming connections with multiple other nodes, thereby creating the multi-hop topology characteristic of a mesh configuration.

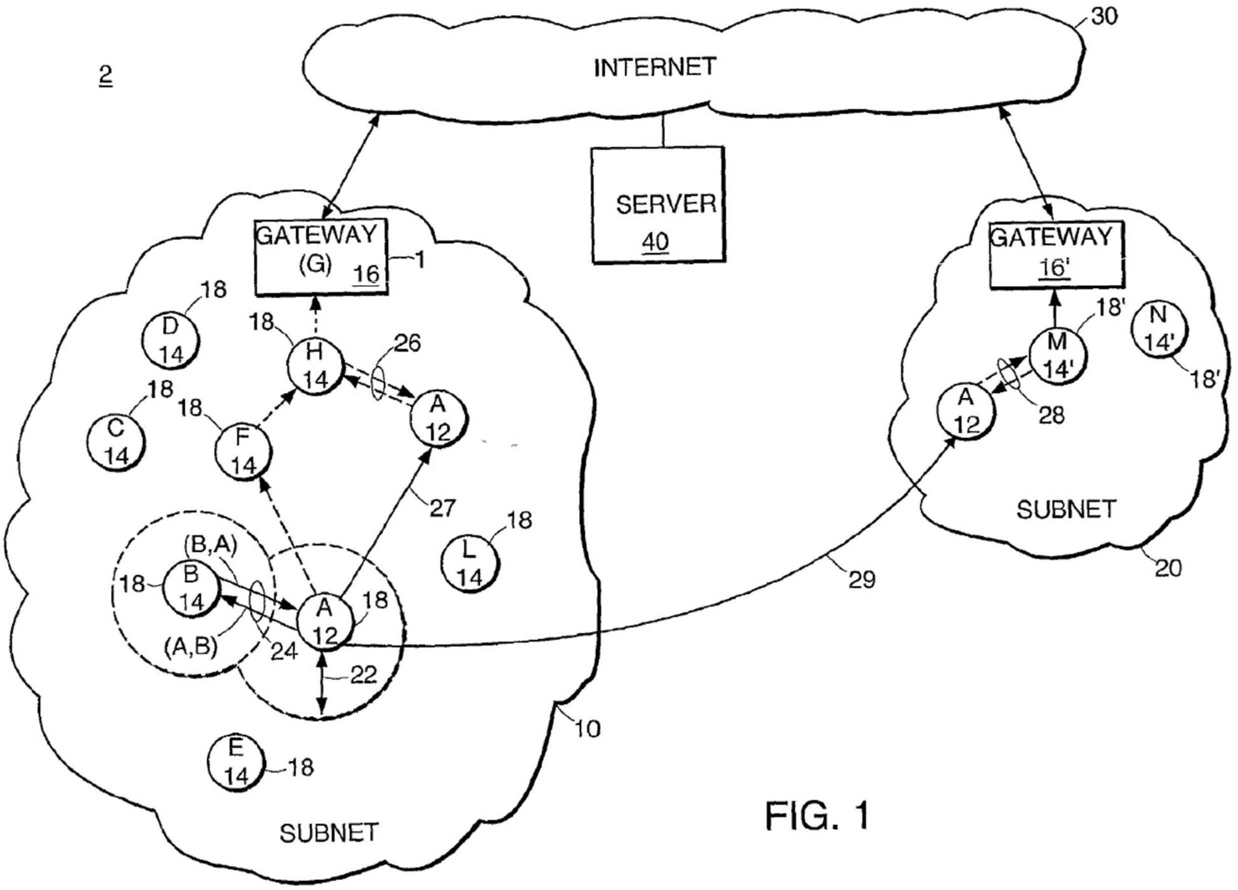


FIG. 1

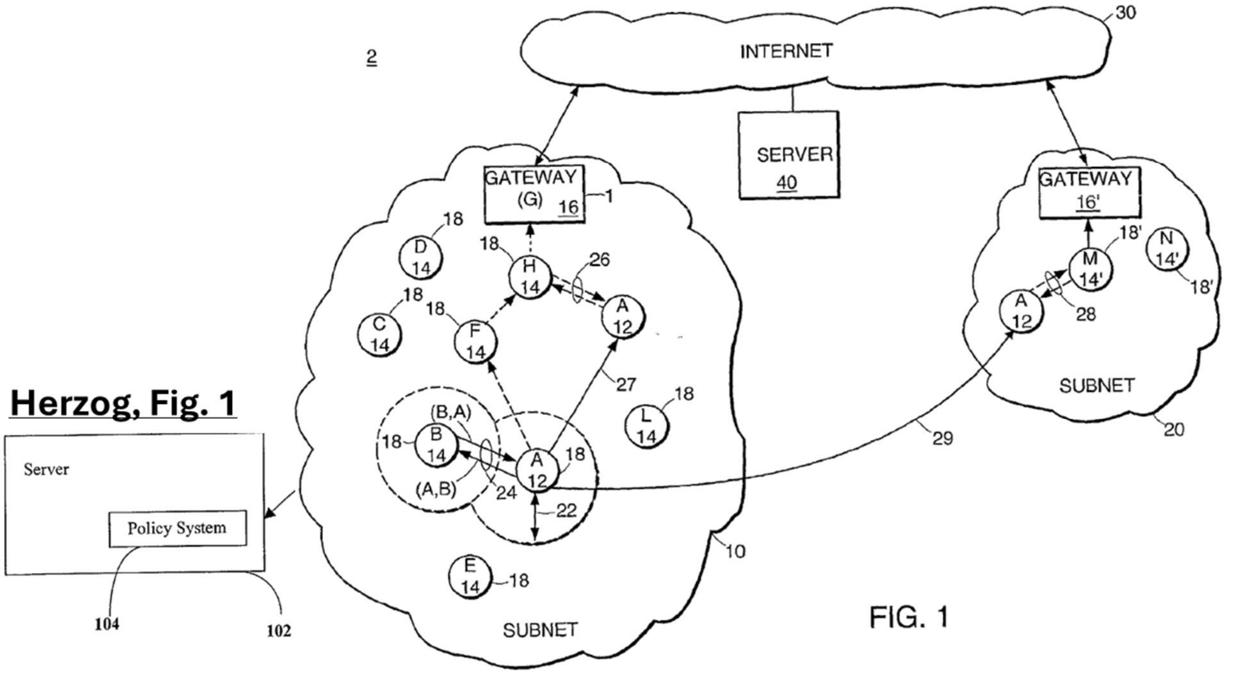
EX1003, Figure 1.

124. ***Second***, Ogier teaches that subnet 10 is a "*wireless*" mesh network. Ogier explains that the nodes 18 within subnet 10 communicate with one another ***wirelessly***. EX1003, [0043] ("In the subnet 10, each node 18 can establish connectivity with one or more other nodes 18 through broadcast or point-to-point links … Such communication links can be … ***wireless***."). This description confirms that Ogier's network is specifically a ***wireless*** mesh network, in which each node can dynamically form wireless broadcast or point-to-point links with other nodes.

b) [9A] an access server wherein the access server sets one or more functioning parameters of the wireless mesh network;

125. Ogier-Shapiro-Herzog teaches [9A].

126. ***First***, the Ogier–Shapiro–Herzog combination teaches "*an access server*." Herzog discloses a "server" that includes a "policy system" which "translates the policy rules into a set of lower-level instructions that network devices understand." EX1005, [0029]. Within the Ogier–Shapiro–Herzog system, those lower-level instructions define the link cost, the goodness factor, and the weight W, as explained in Section VII.A.4 [Ogier–Shapiro–Herzog]. In so doing, Herzog's policy server manages subnet 10 by distributing the link-cost parameters used by Ogier's nodes to compute routing paths, in the same manner the "*access server*" in the '243 Patent "'manages' the network, by setting control parameters for the network." EX1001, 3:34–37 ("Control parameters, set by an access server can then tune the wireless network to provide a mix between the two extremes of max throughput and low latency."); 5:66–67 ("The Access server (10) "manages" the network, by setting control parameters for the network."). Accordingly, a POSITA would have recognized that Herzog's policy server teaches "*an access server*."


Ogier-Shapiro-Herzog Combination

EX1003, Figure 1; EX1005, Figure 1.

127. ***Second***, the policy server in Ogier–Shapiro–Herzog "*sets one or more functioning parameters of the wireless mesh network*." As explained above, Herzog teaches that the server "translates the policy rules into a ***set of lower-level instructions***." EX1005, [0029]. The routing nodes can "understand" said instructions and use them to compute link costs. EX1005, [0029]. Section VII.A.4 [Ogier–Shapiro–Herzog]. In the context of the Ogier–Shapiro–Herzog system described in Section VII.A.4 [Ogier–Shapiro–Herzog], the "set of lower-level instructions"—which include which link cost calculation to use (e.g., LC_1 vs. LC_2), the goodness factor GF, and/or the weight W—correspond to the functioning

parameters of the network. Accordingly, Herzog's policy server sets "*one or more functioning parameters of the wireless mesh network*."

c) [9B] one or more root nodes connected to said access server and an external network;

128. Ogier-Shapiro-Herzog teaches [9B].

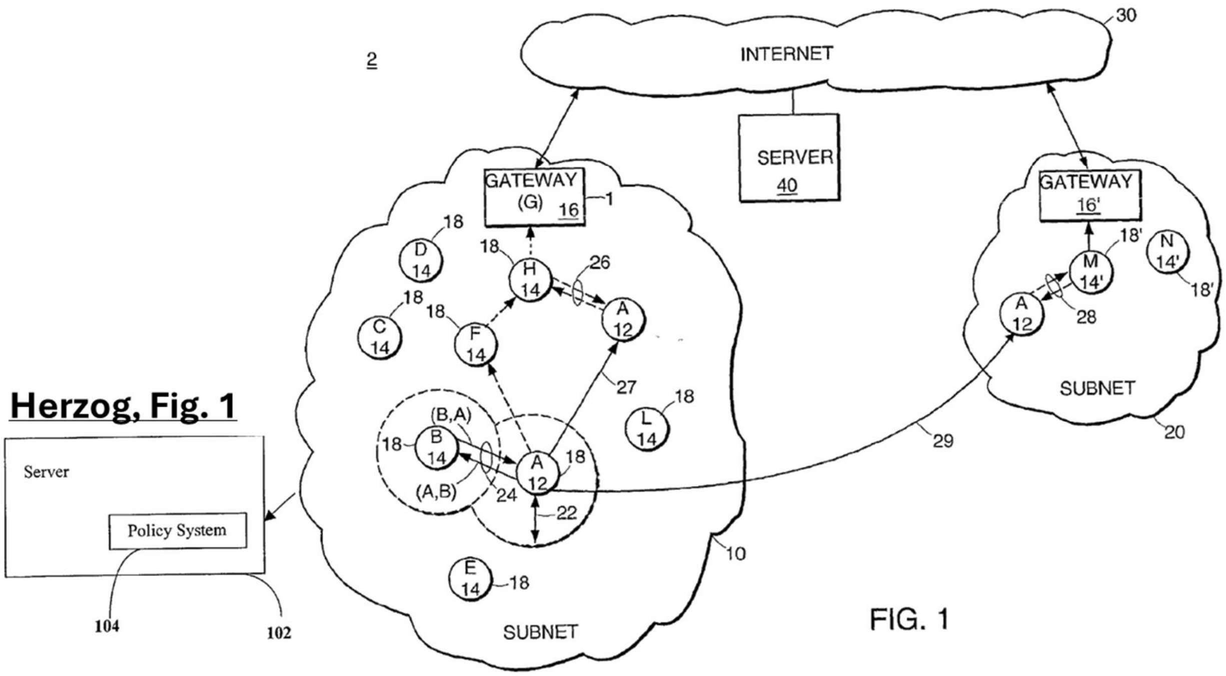
129. ***First***, Ogier discloses "*one or more root nodes*." For example, Ogier teaches that "[t]he subnet 10 includes IP hosts 12, routers 14, and a gateway 16 (collectively referred to as nodes 18)." EX1003, [0040]. It further explains that "[t]he ***gateway 16*** is a particular type of routing node 14 that ***connects the subnet 10 to the Internet 30***." EX1003, [0040]. Ogier also states that, within subnet 10, "[a]ny route taken by packets sent by the IP host A12 to the server 40 on the Internet 30 necessarily traverses IPv4 infrastructure to reach the ***gateway 16***." EX1003, [0261].

130. Since all traffic from clients within the subnet would reach the gateway to communicate with the internet, the gateway serves as the interface between the subnet and external network. Therefore, gateway 16 is the "*root node*" of subnet 10, given that the '243 Patent describes its "root node" as the element that "acts as the interface between the wireless communication devices (30) and the Ethernet." EX1001, 6:15–19.

131. ***Second***, in the Ogier–Shapiro–Herzog combination, Ogier's gateway 16 is "*connected to said access server*." Ogier's gateway 16 connects to and

communicates with Herzog's policy server ("*access server*") through subnet 10. Herzog teaches that its server "is in communication with the [subnet 10] such that the server can communicate with any other devices [i.e., Ogier's nodes, including gateway 16] also connected to the [subnet 10]." EX1005, [0029]. Given that gateway 16 in Ogier's subnet 10 is a within the subnet, it would be connected to Herzog's policy server in the combined system, as depicted below.

Ogier-Shapiro-Herzog Combination



EX1003, Figure 1; EX1005, Figure 1.

132. ***Third***, Ogier's gateway 16 is "*connected to … an external network*." As discussed above Ogier's gateway 16 serves as an interface, and "connects subnet 10 to the Internet 30" ("*external network*"). EX1003, [0040]. *See also* EX1003,

[0261] ("***Any route*** taken by packets sent by the IP host A 12 to the server 40 ***on the Internet 30*** necessarily traverses IPv4 infrastructure to reach the ***gateway 16***."). A POSITA would have understood the internet is an external network.

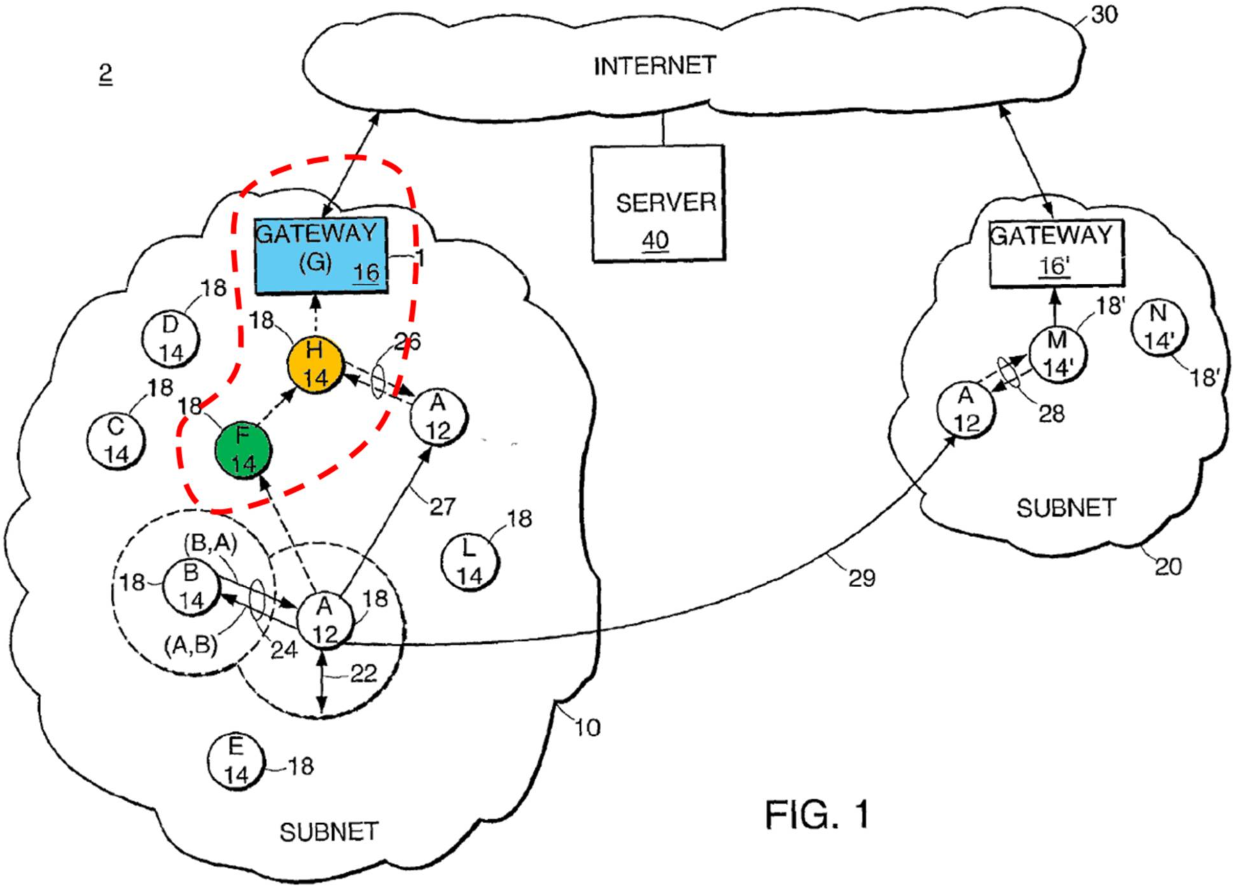
d) [9C] one or more AP nodes wherein each AP node is in wireless two-way data communication with an associated parent node wherein said associated parent node is selected from all available parent nodes wherein an available parent node is another AP node within wireless communication range of the AP node and the associated parent node is an available parent node meeting one or more communication criteria or the associated parent node is a root node within wireless communication range;

133. Ogier-Shapiro-Herzog teaches [9C].

134. ***First***, Ogier discloses "*one or more AP nodes*." Ogier's routers operate as access point ("*AP*") nodes, given that Ogier describes that its subnet 10 includes one or more routers 14, ("*one or more AP nodes*")—node F, for example— "that forward[] IP packets not explicitly addressed to itself." EX1003, [0040]. Ogier's routing nodes each function as an access point, because they provide IP hosts 12 with a point of access to the subnet 10 and, provide access to the Internet 30 through gateway 16 as discussed above.

135. Ogier illustrates this in examples where it discusses nodes communicating while moving in and out of range of the subnet: "[c]onsider, for example, that ***node A [IP host 12]*** is communicating with the server 40 over a route

through subnet 10 that includes the ***link (A, B) to node B [router] 14***." EX1003, [0046]. *See also* EX1003, [0050] ("Upon recovering the same link to node B 14, or upon reestablishing a new link to another node 18 in the same subnet 10 or in the foreign subnet 20, the node A 12 can resume the interrupted communications with Server 40."). Accordingly, Ogier's routers 14 are "*AP nodes*" within the mesh network.



EX1003, Figure 1.

136. ***Second,*** Ogier teaches "*wherein each AP node is in wireless two-way data communication with an associated parent node*." Ogier discloses that "each node 18 [including router 14] can establish connectivity with one or more other nodes 18 through ***broadcast*** … ***links***," which "can be … ***wireless***." EX1003, [0043].

137. Ogier further explains that "[e]ach broadcast link … is mapped into multiple point-to-point ***bi-directional links***." EX1003, [0044]. These bi-directional links are formed between router nodes 14 ("*AP nodes"*). EX1003, [0155] ("neighboring ***router nodes 14*** that have established ***bi-directional links***"); [0200]. Ogier's disclosure of bi-directional wireless links nodes indicates that its routers exchange data wirelessly in both directions. A POSITA would have therefore understood that Ogier teaches "*AP nodes*" that are in "*wireless two-way communication.*"

138. Ogier also teaches that each router 14 ("*each AP node*") has "*an associated parent node*" for communication, because Ogier's neighboring routers operate within a parent–child relationship. For example, Ogier explains that a router 14 chooses a path for routing data to a destination by using a "path selection algorithm" that "appl[ies] Dijkstra's algorithm to compute shortest paths (with respect to cost, c) to [the destination node.]" EX1003, [0198].

139. In carrying out this computation, each router identifies the "parent node[]" (e.g., another router 14) on the selected path toward the destination node and selects it for data forwarding. In other words, the router selects as its "parent node" the next-hop neighbor on that path.

140. As Ogier states: "[w]hen forwarding data packets to a destination node, each routing node 14 selects the ***next node*** on a route to the destination." EX1003, [0059]. This reflects the parent–child relationship between nodes on the path, which was well known in the art. For example, O'Neal describes a routing path as a sequence of parent-child pairs, where each node forwards traffic to its parent until the traffic reaches the root. EX1006, [0067] ("User node A could be thought of as a child node of the server and as a parent node for other user nodes connected directly to it. User node B, a second level user node 13, could be thought of as A's child.").

141. In Ogier, each router 14 functions as an AP node that, once a shortest path is computed, communicates with its parent node (the next hop) to relay data toward the destination.

142. Similarly, as also discussed in Section VII.A.1 [Ogier], a source node distributing link-state information (e.g., link cost) over a "path tree" operates as a root node, with the path tree including parent and child nodes. EX1003, Abstract, [0010] ("Each path tree has the source node as a root node, a parent node, and zero

or more children nodes.”); *see also* EX1003, [0084] (“If node i receives a message representing a link state update … or a change in the cost of a link to a neighbor node, node i … forwards (step 102) the link-state information in a link-state update to the neighbor nodes in children i(src), where src is the source node at which the update originated. Node i then computes (step 104) the parent nodes p i(u) for all potential source nodes Src by running a shortest-path algorithm such as Dijkstra's algorithm.”) ; [0093] (“In brief, the TBRPF protocol disseminates link state updates generated by a source node src along the minimum-hop-path tree rooted at node sc and dynamically updates the minimum-hop-path tree based on the topology and link-state information received along the minimum-hop path tree.”).

143. It should be noted that Ogier uses the term “root node” slightly differently from the ’243 Patent, as Ogier’s “root node” is the node that originates a link state update according to the TBRPF protocol. EX1003, Abstract, [0010] (“Each path tree has that source node as a root node.”). In the ’243 Patent, a “root node” is the node that “acts as the interface between the wireless communication devices (30) and the Ethernet.” EX1001, 6:15–19.

144. However, in the case that Ogier’s gateway 16 serves as the source of a TBRPF link state update, it would qualify as a “root node” under both definitions.

Accordingly, I have conducted my analysis in this petition only considering the case in which the gateway nodes are the source of a link state update.

145. For example, in Figure 1 of Ogier, when the gateway 16 operates as the source node by sending a link-state update, it is the root node of the path tree. Node H (router 14) then becomes the parent node because it is selected as the next hop on the path toward the root, while node F (router 14) becomes the child node of node H.

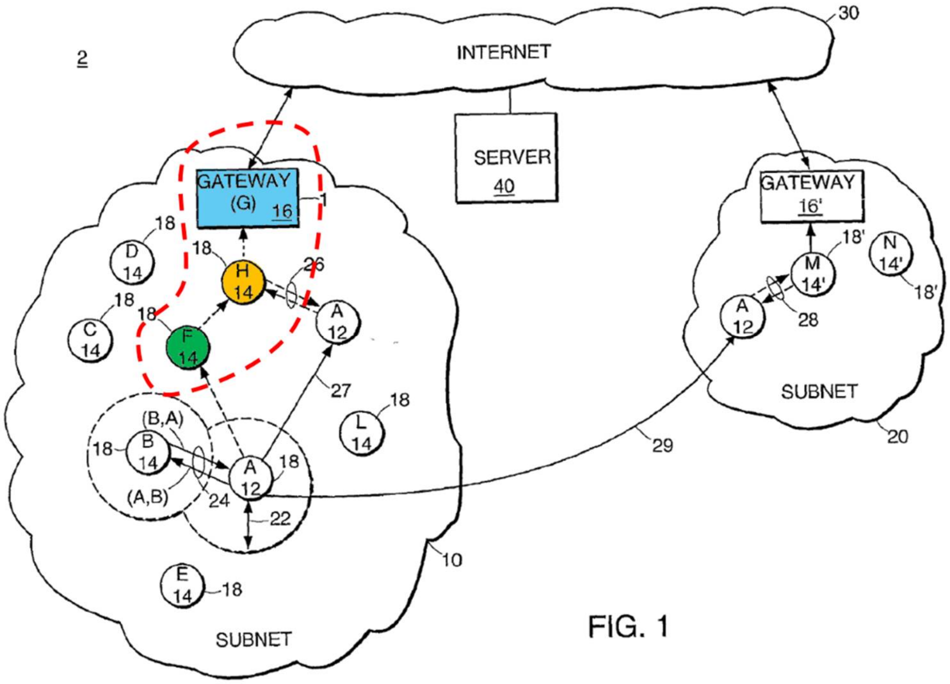


FIG. 1

EX1003, Figure 1.

146. Therefore, Ogier teaches that "*each AP node* [e.g., node F 14] *is in wireless two-way data communication with an associated parent node* [e.g., node H 14]."

147. ***Third,*** Ogier discloses "*wherein said associated parent node is selected from all available parent nodes wherein an available parent node is another AP node within wireless communication range of the AP node*."

148. Ogier teaches that routers determine both the routing path and the TBRPF path tree by selecting a route that minimizes the total link cost. *See* EX1003, [0058] ("embodiments of the TBRPF protocol can use … ***shortest path trees***"); [0084] ("Node i then computes (step 104) the parent nodes p i(u) for all potential source nodes src by running a ***shortest-path algorithm*** such as ***Dijkstra's algorithm***"); [0197]–[0198] ("Routing protocols can also be classified according to whether they find optimal (shortest) routes or sub-optimal routes…. One exemplary path Selection algorithm is to apply Dijkstra's algorithm to compute shortest paths (with respect to cost, c) to all destinations.").

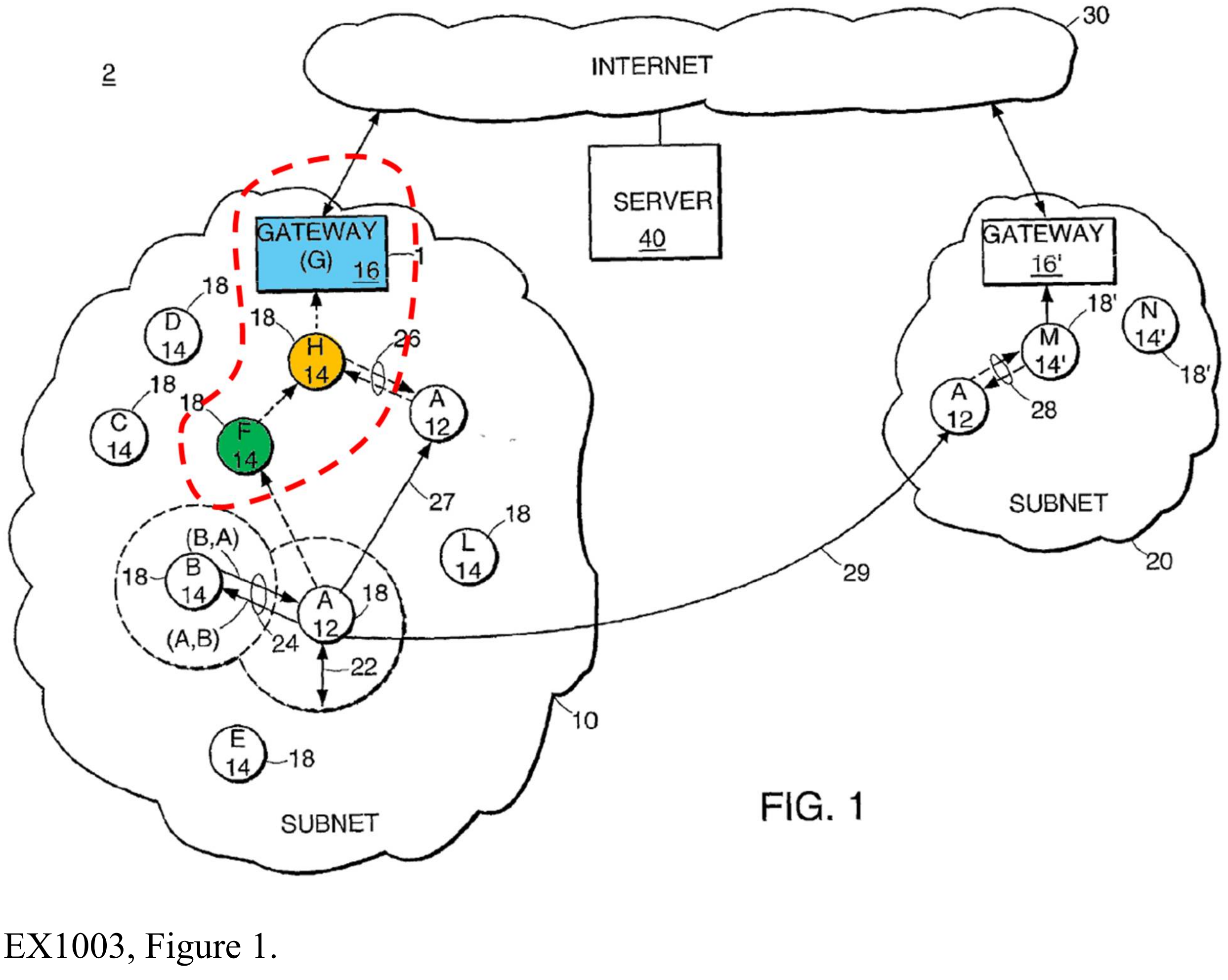
149. Ogier explains that "each routing node 14 has complete link-state information" (EX1003, [0198]) stored in a topology table. EX1003, [0060]–[0061]. Specifically, the "topology table, denoted TT_i, consist[s] of all link-states stored at node i" and maintains the link cost for each pair of nodes u, v in the subnet 10.

EX1003, [0061]. That "link-state information" refers to data describing the status and cost of communication links as discussed in Section VII.A.1, which is used to compute routes in a link-state protocol. Ogier further notes that "each node 14 has link information for every link in the subnet 10." EX1003, [0152].

150. Ogier uses this link-state information both to select the routing path for "transmitting packets" (EX1003, [0196], [0198]) and to build the "shortest path tree" for TBRPF (EX1003, [0058], [0084]). Ogier goes on to teach that "[e]ach wireless node 18 … has a range 22 of communication within which that node 18 can establish a connection to the subnet 10." EX1003, [0043].

151. This range of communication is the physical distance within which two wireless nodes can establish a viable link. Ogier also makes clear that the topology table excludes "failed links (represented by an infinite cost) and links that are unreachable." EX1003, [0101]. Therefore, any nodes out of wireless communication range of the subnet would be excluded from the parent calculation algorithm. As such a particular routing node *i* conducting parent selection only considers routers that are reachable through active wireless links, which teaches that "*said associated parent node is selected from all available parent nodes wherein an available parent node is another AP node within wireless communication range of the AP node*."

152. Ogier illustrates this in Figure 1: when Node F selects an "*associated parent node*" (e.g., Node H), it does so from "*all available parent nodes*" (e.g., Nodes , D, H) that are "*within wireless communication range*" of Node F. *See* EX1003, [0090] ("the parent 124 for node F with respect to source node D is node H"), [0152] (link "(D, F)"); Figure 4 (showing that Node F is within wireless communication range of Node H); Figure 5 (showing that Node F is within wireless communication range of Node D).



EX1003, Figure 1.

153. ***Fourth,*** Ogier-Shapiro-Herzog discloses that "*the associated parent node is an available parent node meeting one or more communication criteria or the associated parent node is a root node within wireless communication range*."

154. For instance, node F in Figure 1 selects its associated parent node H, and node H selects its associated parent node gateway 16, by identifying the neighboring node toward the destination node (gateway 16) along the path that minimizes the total link cost. *See* EX1003, [0058] ("embodiments of the TBRPF protocol can use … ***shortest path trees***"); [0084] ("Node i then computes (step 104) the parent nodes p i(u) for all potential source nodes src by running a ***shortest-path algorithm*** such as ***Dijkstra's algorithm***"); [0197]–[0198] ("One exemplary ***path selection algorithm*** is to apply ***Dijkstra's algorithm*** to compute shortest paths (with respect to cost, c) to all destinations."). Therefore, the "*associated parent node*" is selected based on "*communication criteria*". In this case, the associated parent node H is "*an available parent node meeting one or more communication criteria*," because it lies along the path to the destination node and that path minimizes the total link cost, which in Ogier–Shapiro–Herzog is determined using metrics such as throughput and latency (see Section VII.A.5.h) [9G]). Moreover, because gateway 16 is the "*root node*" in Ogier, "*associated parent node*" [of node H] *is the root node* [the gateway itself], "*within wireless communication range*."

e) [9D] wherein an AP node is in wireless communication with zero or more clients; and

155. Ogier-Shapiro-Herzog teaches [9D].

156. ***First***, Ogier discloses a subnet with "*clients*." As explained in Section VII.A.5.c) [9B], subnet 10 in Ogier includes "IP hosts 12." EX1003, [0040]. Ogier distinguishes IP hosts from routers, explaining: "As used hereafter, a router 14 is any node 18 that forwards IP packets not explicitly addressed to itself, and an IP host 12 is any node 18 that is not a router 14." EX1003, [0040]. Thus, an IP host is a node that does not forward transit traffic but instead receives and processes packets addressed to it. This is the function of a "client" node in a wireless mesh network. Indeed, Ogier itself refers to node 12 as "client 12." EX1003, [0335]–[0336].

157. ***Second***, Ogier teaches that a router 14 ("*AP node*") "*is in wireless communication with zero or more*" IP hosts 12 ("*clients*"). For example, Ogier discloses that all of its nodes 18 within the subnet—including both routers 14 and IP hosts 12—may be "wireless," and that a router 14 may be in wireless communication with an IP host A 12, as shown in Figure 1. EX1003, [0043] ("[N]odes 18 are referred to as wireless…"; "Each wireless node 18 …"); [0040]. Figure 1 illustrates this arrangement: IP host A (node 12) communicates wirelessly with router B (node 14), with their wireless ranges represented by larger concentric circles.

158. Ogier also demonstrates that the association between routers (AP nodes) and IP hosts (clients) may vary. As shown in Figure 1, a router 14 may be in communication with "*zero*" IP hosts (such as router D), or with "*more*" than one IP host (such as router F). Because routing nodes serve as points of access for client nodes, the number of clients connected to a given AP node depends on network topology, traffic demands, and wireless range.

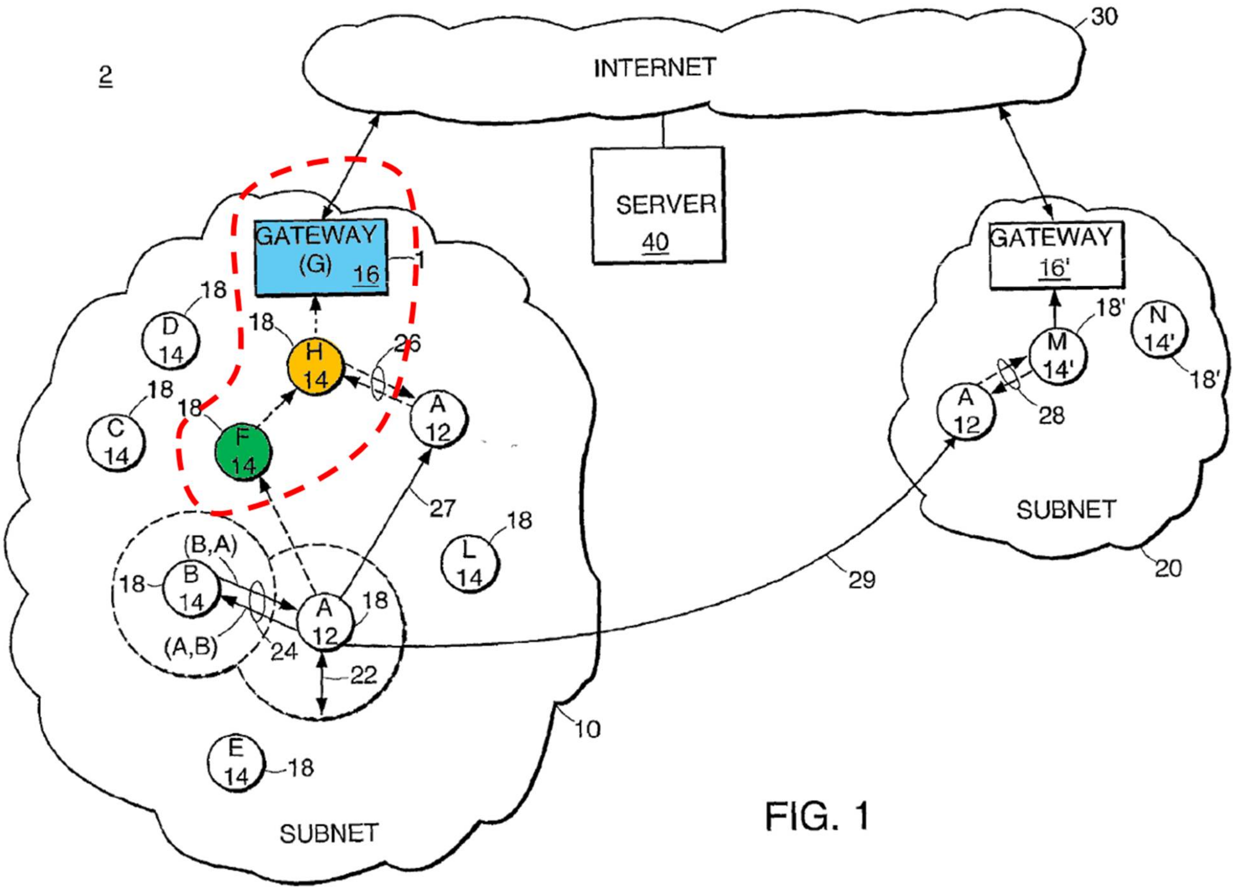


FIG. 1

EX1003, Figure 1.

f) [9E]

a. [9E-1] wherein an AP node includes a means for switching two-way data communication from a first associated parent node to a second associated parent node based on the functioning parameters of the wireless mesh network and

159. As explained in Section VI.A.1, "a means for switching two-way data communication from a first associated parent node to a second associated parent node based on the functioning parameters of the wireless mesh network" should be interpreted under 35 U.S.C. 112 ¶6 to require the following three: (1) *a processor and a storage medium with instructions that* (2) *perform the identified function* (3) *based on latency and throughput*.

160. Ogier-Shapiro-Herzog teaches or renders obvious [9E-1] under such construction.

161. (1): Ogier teaches a router that includes a processor and a storage medium with instructions.

162. A POSITA would have understood that routers, as a matter of standard design, are implemented using a processor to execute software instructions stored in memory. This was well known in the art. *See* EX1010 [Inouchi], 9:46–49; 13:29–32; 13:33–41 ("The ***routing controller*** 50 comprises a ***processor*** 20, a ***program***

memory 21"); 13:43–44 ("The ***program memory*** 21 contains an OS 211 and a variety of ***programs to be carried out by the processor 20***.").

163. Ogier's routers implement their routing functions—including maintaining a topology table and computing paths—by means of a processor executing instructions stored on a storage medium. Accordingly, Ogier teaches the structural components required under the §112 ¶6 construction for "a processor and a storage medium with instructions."

164. (2): Ogier-Shapiro-Herzog teaches that the router 14 performs "*switching two-way data communication from a first associated parent node to a second associated parent node based on the functioning parameters of the wireless mesh network*." Ogier teaches "*switching two-way data communication from a first associated parent node to a second associated parent node*" for two separate reasons.

165. ***First***, Ogier identifies that its router 14 selects a routing path using "a path selection algorithm" that minimizes the total link cost. EX1003, [0198]. Link costs "can vary in time" (EX1003, [0044]) , therefore, each router "update[s] [the preferred] paths when link states are updated." EX1003, [0198]. In Ogier's system, when link-state updates occur, the router recomputes its shortest path tree using the updated link costs. As explained in Section VII.A.5.d) [9C], such recomputation

results in selecting a new parent node along the updated path. When that occurs, the router switches its forwarding relationship from the previous parent node to the newly selected parent node. Ogier describes this process in terms of updating "bi-directional links" ("*two-way data communication*") between routers. EX1003, [0044].

166. ***Second***, Ogier also discloses that a router can update the "shortest path tree" for TBRPF (EX1003, [0058]) by "run[ning] a shortest-path algorithm such as Dijkstra's algorithm" (EX1003, [0084]) on the topology table TT_i. When the recomputation results in a change of parent, Ogier teaches: "If this computation results in a change to the parent node p_i(u) for any source u, node i then sends a NEW PARENT(u, sn) message, where sn=sn_i(u), to the new parent node p_i(u) ["*second associated parent node*"] and a CANCEL PARENT message to the old parent node ["*first associated parent node*"] (step 106)." EX1003, [0084].

167. Sending the NEW PARENT message formally establishes the new parent–child relationship in the path tree, while the CANCEL PARENT message terminates the prior relationship. Therefore, this process effects the switching of bi-directional links ("*two-way data communication*") (EX1003, [0044]) from the first associated parent node to the second associated parent node. Accordingly, Ogier

teaches the claimed function of switching two-way data communication between parent nodes.

168. Ogier-Shapiro-Herzog also teaches that parent-node switching is "*based on the functioning parameters of the wireless mesh network*." As explained in Section VII.A.4 [Ogier-Shapiro-Herzog], link costs are computed with the following two formulas:

a. $LC_1 = 1 + W*GF$; or

b. $LC_2 = (LD + bias) + W*GF$.

169. As discussed in Section VII.A.4 [Ogier-Shapiro-Herzog] and Section VII.A.5.b) [9A], in the Ogier–Shapiro–Herzog system, Herzog's server provides each node a "set of lower-level instructions"—including the link-cost definition (LC_1 vs. LC_2), the goodness factor (GF), and/or the weight (W)—which each node then implements into its link-cost calculations to comply with the network wide link-cost policy set by the network administrator. EX1005, [0005] ("Network devices understand the instructions, and the specified goals thus can be accomplished."). The "lower-level instructions" constitute the "*functioning parameters of the wireless mesh network*".

170. Accordingly, when a node recomputes its routing path and determines whether to switch from one parent node to another, it selects a new routing path or

path tree—and by extension, a new associated parent node—on the basis of the "lower-level instructions" ("*functioning parameters of the wireless mesh network*") distributed by Herzog's policy server.

171. (3): Ogier-Shapiro-Herzog also teaches that the parent-node is conducted "*based on latency and throughput*."

172. As explained above, a node switches parent nodes based on the "lower-level instructions" defining the link-cost computation—specifically, whether to apply LC_1 or LC_2, how the goodness factor (GF) is defined, and the value of the weight (W). Section VII.A.4 [Ogier–Shapiro–Herzog]. Ogier teaches that LC_1 favors a "minimum-hop route," while LC_2 favors a route with smaller "***link delay***." EX1003, [0044]. Shapiro teaches that the GF reflects "a decaying average of periodically sampled ***throughput*** for a node." EX1004, [0048].

173. A POSITA would have understood that when router 14 applies these instructions, it executes software that switches parent nodes based on the link delay (latency) and throughput, as encompassed by the link-cost function. Link delay, as taught by Ogier, would have been understood to be synonymous with latency in this context.

174. The ’243 Patent also uses the terms interchangeably when referring to voice transmission application requiring “low ***delay (latency)***.” EX1001, 1:51–53. That was common in the art. EX1011, 11:32-36.

175. Thus, the Ogier–Shapiro–Herzog combination discloses instructions executed by a processor to switch parent nodes based on link delay (“*latency*”) and throughput, for example, when using LC_2.

b. [9E-2] wherein an AP node contains one or more datasets

176. Ogier-Shapiro-Herzog teaches “*wherein an AP node contains one or more datasets*” because a router includes a “*route path dataset*” (*see* Section VII.A.5.g) [9F]), and “*a dataset of child node identifiers*” (*see* Section VII.A.5.i) [9H]).

g) [9F] wherein one of the datasets contained in an AP node comprises a route path dataset comprising an identifier for the associated parent node appended to the route path dataset for the associated parent node;

177. Ogier-Shapiro-Herzog teaches [9F].

178. During prosecution, Applicant introduced limitation [9F] as claim 3 of the Preliminary Amendment filed on December 8, 2009. EX1008, 114. In doing so, Applicant explained that “[s]pecification ¶45 and Figure 4 provide that each network node contains a ***connection path dataset (the ‘RP: Route Path’) as is claimed in***

new claim 3." EX1008, 118. Paragraph 45 of the originally submitted specification states, in relevant part: "RP: Route Path. In FIG. 6(1), the route path for 005 is 000 002 005, its connection route." EX1008, 12. Figures 4 and 6 of the originally submitted application, reproduced below, further illustrate this disclosure.

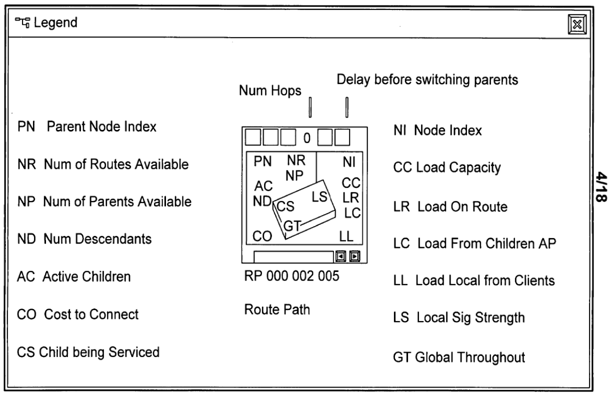


FIG. 4

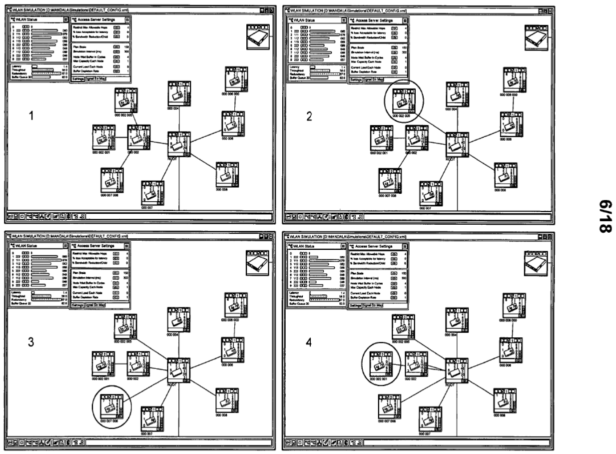


FIG. 6

EX1008, 49, 51.

179. Patent Owner has also taken a similar position in its Infringement Contentions for the '243 Patent. Specifically, Patent Owner claims a "dataset comprising 'the best path back to a RAP[root AP]'" constitutes a "*route path dataset*." EX1014, 21.

180. If paragraph 45 and Figure 4 of the originally submitted application provide sufficient written-description support for the claimed "*route path dataset,*" Ogier-Shapiro-Herzog also teaches [9F].

181. Ogier teaches that "[e]ach routing node 14 … compute[s] preferred paths to all possible destinations, and to update these paths when link states are updated." EX1003, [0198]. Ogier further explains: "Once preferred paths are computed, the routing table entry for node u is set to the next node on the preferred path to node u." EX1003, [0198].

182. That "preferred path" includes a sequence of identifiers of nodes leading from the source to the destination, as reflected in Ogier's disclosures. For example, Ogier describes "using as path name the ***sequence of nodes in the path***" (EX1003, [0145]) and that "[t]he routing table entry for node u, consisting of the next node on a preferred path to node u." EX1003, [0067]. Ogier further discloses packet forwarding "through routers along the path…." EX1003, [0335]. Ogier's representation of a "path" as a sequence of node identifiers was common in the art. For example, as shown by O'Neal, a path was typically represented as a sequence of node identifiers: "***Path D-B-A-S*** (where 'S' is the server) represents the shortest available path." EX1006, [0073].

183. Accordingly, the Ogier–Shapiro–Herzog system teaches the claimed "*route path dataset*," since each node computes, stores, and uses paths including sequences of node identifiers to reach a given destination.

h) [9G] wherein the communication criteria comprises instructions for the AP node to select the associated parent node wherein an available parent node is selected to become the associated parent node if the available parent node is in wireless communication with a root node or if a root node is contained in the available parent node's route path dataset; and

184. Ogier-Shapiro-Herzog teaches [9G].

185. ***First***, Ogier-Shapiro-Herzog teaches "*wherein the communication criteria comprises instructions for the AP node to select the associated parent node*."

186. Claim 9 appears to use the claim term "*communication criteria*" in two ways. For example, limitation [9C] appears to suggest that the "communication criteria" comprise "criteria" that can be "me[]t" by a node, while [9G] appears to suggest that the communication criteria comprises "***instructions*** … to select" a node based on criteria. Regardless of whether the "communication criteria" comprises "criteria" or "instructions," Ogier-Shapiro-Herzog teaches the claimed "communication criteria" because a router selects its associated parent node by identifying the node that satisfies certain criteria, and as explained in Section VII.A.5.f) [9E], the router implements its functionalities using "*instructions*" executed by a processor.

187. For example, as explained in Section VII.A.5.d) [9C], node F in Figure 1 selects its associated parent node (H) by evaluating its neighboring nodes (i.e.,

available parent nodes) and identifying the one that: (1) lies along a path toward the destination node, and (2) is on the path that minimizes the total link cost. *See* EX1003, [0058] ("embodiments of the TBRPF protocol can use … ***shortest path trees***"); [0084] ("Node i then computes (step 104) the parent nodes p i(u) for all potential source nodes src by running a ***shortest-path algorithm*** such as ***Dijkstra's algorithm***"); [0197]–[0198] ("One exemplary ***path selection algorithm*** is to apply ***Dijkstra's algorithm*** to compute shortest paths (with respect to cost, c) to all destinations.").

188. Accordingly, "*the communication criteria comprises instructions for the AP node* [e.g., node F] *to select the associated parent node* [node H]" that comports with those criteria, exactly as described in Ogier's path-selection process.

189. ***Second***, Ogier teaches "*wherein an available parent node is selected to become the associated parent node if the available parent node is in wireless communication with a root node or if a root node is contained in the available parent node's route path dataset.*"

190. As explained in Section VII.A.1[Ogier], a router 14 (e.g., node F) can determine a path to gateway G 16 by "comput[ing] shortest paths (with respect to cost, c) to" the gateway G 16. EX1003, [0198]. *See also* EX1003, [0084] ("running

a shortest-path algorithm such as Dijkstra's algorithm"). The resulting shortest path between node F and gateway G 16 in Figure 1 is illustrated below (in purple).

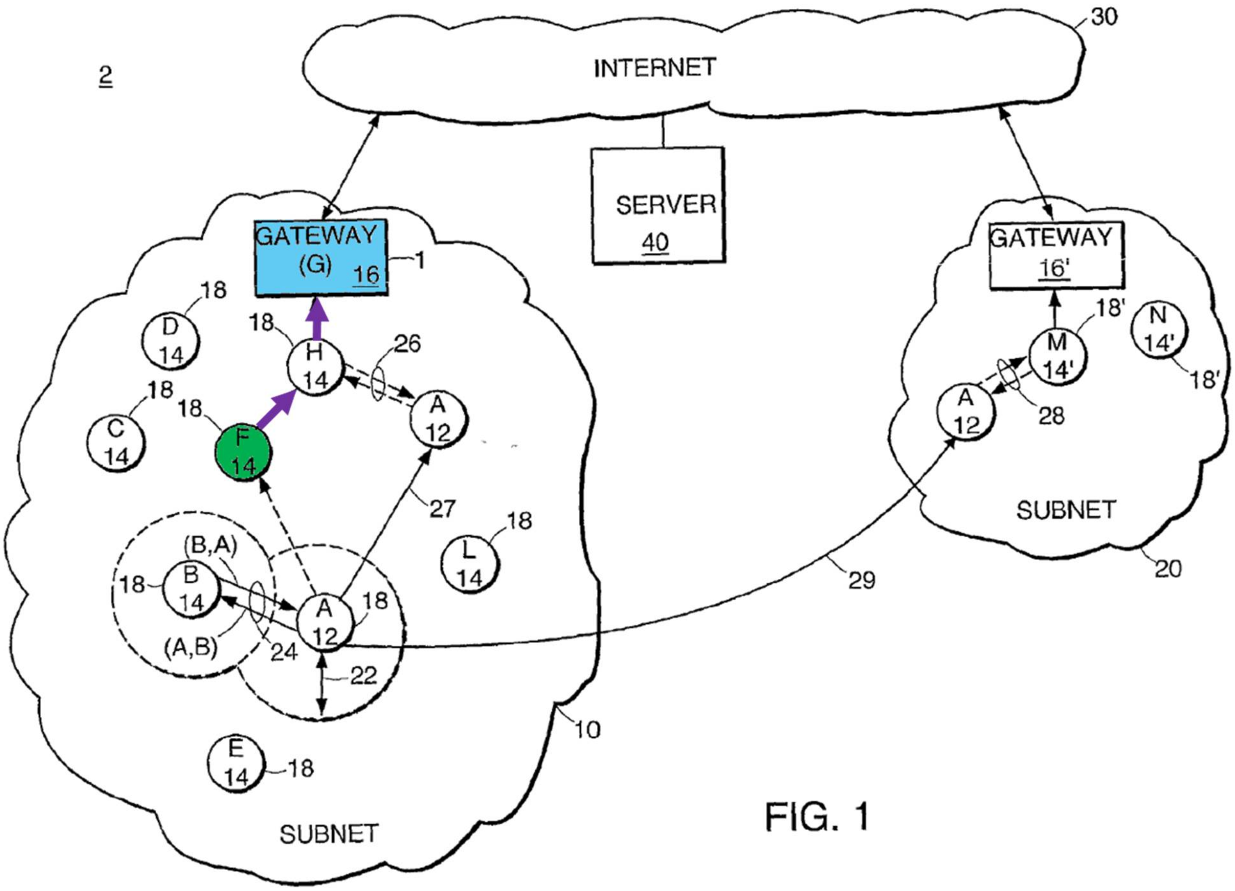


FIG. 1

EX1003, Figure 1.

191. In this example, node F selects node H as its "associated parent node" because node H is "in wireless communication with a root node" (i.e., gateway G 16). EX1003, [0090]. Ogier further emphasizes the central role of the gateway, explaining: "Any route taken by packets sent by the IP host A 12 to the server 40 on

the Internet 30 necessarily traverses IPv4 infrastructure to reach the gateway 16." EX1003, [0261].

192. Since node H lies on the routing path to gateway G 16 (the "root node"), node H's routing path dataset includes the identifier of the gateway. EX1003, [0198] ("Each routing node 14 then applies a path Selection algorithm to compute preferred paths to all possible destinations, and to update these paths when link states are updated."). This interpretation is reinforced by Ogier's description of "using as path name the sequence of nodes in the path" (EX1003, [0145]), its definition of the routing table entry as identifying the next node on a preferred path (EX1003, [0067]), and its reference to forwarding packets "through routers along the path" (EX1003, [0335]). See also EX1006, [0073] ("Path D-B-A-S").

i) [9H] wherein one of the datasets contained in an AP node comprises a dataset of child node identifiers wherein the dataset of child node identifiers is a dataset identifying each AP node in wireless communication with the AP.

193. Ogier-Shapiro-Herzog teaches [9H].

194. During prosecution, Applicant introduced [9H] as claim 9 of the Preliminary Amendment filed on December 8, 2009. EX1008, 115. In reference to the newly introduced amendment, Applicant stated that "[t]he specification ¶84 also teaches that each node will be aware of the children that are connected to the node,

as claimed by claim 9." EX1008, 119. *See also* EX1008, 119 ("the mesh node must be aware of all active children as otherwise the mesh node could not service each of the child nodes.").

195. If paragraph 84 of the originally submitted application provides sufficient written description support for [9H], Ogier-Shapiro-Herzog also teaches [9H].

196. Ogier teaches that each node *i* maintains, "[f]or each node u other than node i," "***[a] list of children nodes*** of node i, denoted children_i(u)." EX1003, [0065]; *see also* [0085]. Ogier defines this list as "the set of neighbor nodes from which node i has received a NEW PARENT message containing the identity of source node [u] without receiving a subsequent CANCEL PARENT message for that source node [u]." EX1003, [0088]. Thus, each node dynamically tracks its direct downstream nodes thereby maintaining an up-to-date record of its active children.

197. Furthermore, Ogier teaches that the list of children nodes includes ***all active*** children. For example, "[i]n one embodiment, the link-state update passes to ***every child node*** in children_i" so that all children are serviced. EX1003, [0111]. Ogier also provides a variation: "those nodes having ***only one child node*** for the source node [u] can send updates … ***to that child node only***." EX1003, [0113].

198. Thus, the referenced list of "children" encompasses the set of downstream nodes that rely on a given parent node for connectivity within the routing tree. By maintaining a list of such child nodes and providing that updates are sent to them, Ogier discloses teaches a system in which each node is aware of its active children.

199. Ogier further teaches that each pair of neighboring router nodes (e.g., node i and its active children) communicate over wireless bidirectional links. *See* EX1003, [0043]–[0044] ("In the subnet 10, each node 18 can establish connectivity with one or more other nodes 18 through … wireless communication links … Each wireless node 18, e.g., IP host A12, … can establish a connection … with other nodes 18 in the subnet 10."). *See also* Section VII.A.5.d) [9C].

6. Claim 12

200. Claim 12 is identical to claim 9 except for limitation [12H]. The combination of Ogier, Shapiro, and Herzog renders obvious claim limitations [12Pre]–[12G] for the reasons provided above for claim limitations [9Pre]–[9G], respectively.

a) [12H] wherein the communication criteria further comprises instructions for the AP node to associate with a single suitable parent node wherein the route path dataset of the parent node is the shortest route path dataset of all available parent nodes.

201. During prosecution, Applicant submitted [12H] as claim 12 of the Preliminary Amendment filed on December 8, 2009. EX1008, 116. In reference to the newly introduced amendment, Applicant stated that "[c]laim 12 covers selection of a parent for association based on the ***shortest route path***, which is discussed in the Specification ¶¶51 and 55 which discuss that a minimal latency is achieved by selecting shortest paths to the root." EX1008, 119.

202. At least under this understanding of the claim, Ogier-Shapiro-Herzog teaches [12H].

203. ***First***, Ogier teaches "*wherein the communication criteria further comprises instructions for the AP node to associate with a single suitable parent node*." In the Ogier–Shapiro–Herzog combination, a router (i.e., an "AP node") chooses a path tree or a routing path "by running a shortest-path algorithm such as Dijkstra's algorithm." EX1003, [0084]; *see also* EX1003, [0198]("Each routing node 14 then applies a path selection algorithm to compute preferred paths to all possible destinations, and to update these paths when link states are updated."); Sections VII.A.5.d) and VII.A.5.f) [9C and 9E].

204. The purpose of running Dijkstra's algorithm on the link costs is to construct a routing path (or a path tree) and, for each destination, to identify the suitable next-hop neighbor. In doing so, the router selects a single parent node from among its available neighbors. Ogier confirms this process: "When forwarding data packets to a destination node, each routing node 14 selects the **next node** on a route to the destination." EX1003, [0059]. Ogier further specifies: "The parent, denoted p_i(u), which is the **neighbor node** ('nbr') of node i that is the next node on a [] path from node i to node u." EX1003, [0064].

205. The singular tense referring to the "parent node" in this context indicates that a ***single*** upstream neighbor is chosen by the AP node as the next hop toward the destination in the shortest path tree. By applying Dijkstra's algorithm, each AP node selects exactly one such parent. Thus, the Ogier–Shapiro–Herzog system teaches the claimed requirement that each AP node selects a single suitable parent node based on the computed path.

206. ***Second***, Ogier teaches that the "*single suitable parent node*" is associated with the "*AP node*" "*wherein the route path dataset of the parent node is the shortest route path dataset of all available parent nodes*." Because the router selects the routing path or path tree by running a shortest-path algorithm (EX1003, [0084] ("running a shortest-path algorithm such as Dijkstra's algorithm"); [0198]

("[e]ach routing node 14 then applies a path selection algorithm to compute preferred paths to all possible destinations")), the route path associated with the selected parent node is the shortest route path among all available parent nodes. *See* EX1003, [0047] ("The link-state-routing protocol, referred to as a topology broadcast based on reverse-path forwarding (TBRPF) protocol, seeks to substantially minimize the amount of update and control traffic required to maintain shortest (or nearly shortest) paths to all destinations in the subnet."); [0197] ("routing protocols can also be classified according to whether they find optimal (shortest) routes or sub-optimal routes."); Sections VII.A.5.d) and VII.A.5.f) [9C and 9E].

207. When Dijkstra's algorithm is applied over the link costs in the topology table, it finds the lowest-cost path to the destination. Thus, when an AP node selects a parent node, the route path corresponding to that parent is the shortest available route path in terms of the defined cost metric (whether hop count, link delay, or a cost incorporating throughput via Shapiro's goodness factor). Accordingly, the Ogier–Shapiro–Herzog system teaches that the selected parent node is chosen because its route path is the shortest among all available parent nodes.

208. If the "*shortest route path dataset*" is taken to mean the route path with the minimum number of hops to gateway 16 ("*root node*")—which is Patent Owner's interpretation of "*shortest route path dataset*" given its Infringement

Contentions (*see* EX1014, 53-55 (mapping "*shortest route path dataset*" to "route path dataset with the lowest number of hops back to the RAP)")—the Ogier-Shapiro-Herzog combination also teaches this. For example, when the nodes are instructed by Herzog's policy server to use link cost as LC_1 with W=0 (see Section VII.A.4.a), running a "shortest-path algorithm such as Dijkstra's algorithm" would yield the minimum-hop path. *See* EX1003, [0044] ("the cost of a link can be one, for minimum-hop routing.").

209. In this configuration, the functioning parameter supplied by the policy server directs Ogier's routers to treat each link as having equal cost. Using Dijkstra's algorithm with said inputs, the computed route path corresponds to the path with the fewest number of hops between the source node and the destination (e.g., the root node).

7. Claim 13

210. Claim 13 is identical to claim 9 except for limitation [13H]. The combination of Ogier, Shapiro, and Herzog renders obvious claim limitations [13Pre]–[13G] for the reasons provided above for claim limitations [9Pre]–[9G], respectively.

a) [13H] wherein the access server functioning parameters includes a latency modifier wherein the AP node means for switching from the first associated parent node to a second associated parent node result in selection of the second associated parent wherein the route path of the second associated parent node is shorter than the first associated route path by a value related to the latency modifier.

211. During prosecution, Applicant introduced [13H] as claim 14 of the Preliminary Amendment filed on December 8, 2009. EX1008, 116. In reference to the newly introduced amendment, Applicant stated that "[t]he use of the access server to increase a ***latency modifier*** thereby causing some of the mesh nodes to switch parents, ***covered by claim 14***, is discussed in ¶66 ('The access server can force this by increasing the ***latency cost factor***, resulting in nodes that can connect to the root directly to do so ….')." EX1008, 119.

212. At least under this understanding of the claim, Ogier-Shapiro-Herzog teaches [13H].

213. ***First***, Ogier-Shapiro-Herzog teaches "*wherein the access server functioning parameters includes a latency modifier*." As discussed in Section VII.A.5.b) [9A], the "functioning parameters of the wireless mesh network" are: (1) the link-cost definition (LC_1 vs. LC_2); (2) the goodness factor (GF) definition; and/or (3) the weight (W). When Herzog's policy server configures the routers to use LC_2, the weight W controls the relative contribution of the goodness factor (GF)

compared to the link delay (LD) in the cost calculation. *See* Section VII.A.4.a) [Ogier–Shapiro–Herzog] ("LC_2 = (LD + bias) + W*GF"); *see also* EX1004, [0053] ("Efficiency=K(Hops)+goodness factor.").

214. In the combination, varying the value of W controls the relative impact of latency vs. throughput. For example, a small W reduces the impact of GF (throughput) and increases the impact of LD (latency) in the cost computation, while a larger W increases the weight of GF (throughput) and reduces the impact of LD (latency) in the routing decision.

215. Given that W is a value that Herzog's policy server supplies nodes that adjusts how latency is prioritized relative to throughput, a POSITA would have recognized that W is a "functioning parameter[]" of the wireless mesh network and specifically operates as a "latency modifier."

216. ***Second***, Ogier-Shapiro-Herzog teaches "*wherein the AP node means for switching from the first associated parent node to a second associated parent node result in selection of the second associated parent*." As explained in Section VII.A.5.f)a [9E-1], the "*means for switching*" selects and switches to the "*second associated parent*."

217. ***Third***, Ogier-Shapiro-Herzog teaches "*wherein the route path of the second associated parent node is shorter than the first associated route path by a value related to the latency modifier*."

218. In the Ogier–Shapiro–Herzog combination, a router determines its routing path or path tree "by running a shortest-path algorithm" on the total link costs. EX1003, [0047] ("The link-state-routing protocol, referred to as a topology broadcast based on reverse-path forwarding (TBRPF) protocol, seeks to substantially minimize the amount of update and control traffic required to maintain shortest (or nearly shortest) paths to all destinations in the subnet."), [0084] ("running a shortest-path algorithm such as Dijkstra's algorithm"), [0197]–[0198] ("routing protocols can also be classified according to whether they find optimal (shortest) routes or sub-optimal routes."); see also Sections VII.A.5.d) and VII.A.5.f) [9C and 9E].

219. The shortest-path algorithm compares the total costs of candidate paths and selects the one with the lowest value. When link-cost computations incorporate both latency (LD) and throughput (GF) with weighting factor W, the selection of a new parent node occurs because the second associated path is shorter—i.e., lower in cost—than the first, by an amount determined by the functioning parameter W.

220. Accordingly, when LC_2 is used, the route path of the second associated parent node is shorter than the first associated route path in terms of the composite cost function LC_2. Because LC_2 is defined as (LD + bias) + W*GF, the relative weighting factor W influences how much throughput (GF) contributes compared to latency (LD).

221. Thus, the route path of the "*second associated parent node*" would be "*shorter*" than that of the "*first associated parent node*" "*by a value related to*" W, which in this context operates as a "*latency modifier*."

B. Ground II: Ogier In View Of Shapiro, Herzog, And Inouchi Renders Obvious Claims 9, 12, And 13.

222. If Patent Owner argues that Ogier-Shapiro-Herzog in Ground I does not render Claims 9, 12, and 13 obvious because it does not explicitly identify "*a processor and a storage medium with instructions*" as required under the proposed construction of "*means for switching*" (limitation [9E]), it would have been obvious to include such components in view of Inouchi.

223. Inouchi teaches a router with a "routing controller 50 [that] comprises a processor 20, [and] a program memory 21." EX1010 [Inouchi], 9:47–49 ("The node apparatus 1A is referred to as an origination ***router***"); 13:29–32 ("The node apparatus comprises … a ***routing controller*** 50"); 13:32–41 ("The routing controller 50 comprises a ***processor*** 20, ***a program memory*** 21"). Inouchi further explains that

"T[t]he ***program memory*** 21 contains an OS 211 and a variety of programs to be ***carried out by the processor*** 20." EX1010, 13:43–45.

224. A POSITA would have understood that this disclosure confirms the well-known physical components of routers at the time of the alleged invention: routers were implemented using a processor that executes instructions stored in memory, including an operating system and routing software.

225. It would have been obvious to modify Ogier's router to include a storage medium with instructions and a processor, as disclosed in Inouchi. The proposed modification merely combines prior art elements: (1) the router in Ogier–Shapiro–Herzog, which performs path computation and parent-node selection, and (2) the program memory and processor in Inouchi's router, which execute operating system and routing instructions. EX1010, 9:47–49; 13:29–45. A POSITA would have recognized that combining these teachings requires nothing more than using conventional electrical connections to provide a processor and memory within Ogier's router. The result would have been predictable: the processor and program memory execute the routing processes described in Ogier–Shapiro–Herzog.

226. A POSITA would have understood this outcome as routine, since it was common in the art to implement routers with processors and memory units running stored programs. The combination therefore does not alter the underlying

functionality of Ogier's routers; it simply reflects the common implementation detail described by Inouchi. Thus, the modification would have been obvious to a POSITA.

C. Ground III: Ground I or II, Further In View Of O'Neal Renders Obvious Claims 10 And 11.

1. O'Neal (EX1006)

227. O'Neal discloses a system that "arrang[es] nodes for distribution of data over a computer network." EX1006, [0002]. O'Neal, like Ogier teaches that each node maintains a topology database (EX1006, [0030])—illustrated in Figure 13—that includes a "sibling database" 134. EX1006, [0165]. To maintain this sibling database, "a parent node reports to each of its child nodes the addresses of their siblings" (EX1006, [0120]), and "the child node stores information about its sibling (or siblings) in the sibling portion (or sibling database) 134." EX1006, [0165].

228. O'Neal refers to child nodes that share the same parent as "siblings." O'Neal teaches this relationship: in Figure 23, "nodes Q and X know that they are each other's ***siblings*** and nodes A and B know that they are each other's ***siblings***." EX1006, [0165].

229. By having parent nodes report sibling information and having child nodes maintain that information in a sibling database, O'Neal discloses that each node is aware not only of its parent and children but also of its sibling peers. A

POSITA would have recognized that this facilitates coordination among children of the same parent—for example by enabling sibling-based communication paths.

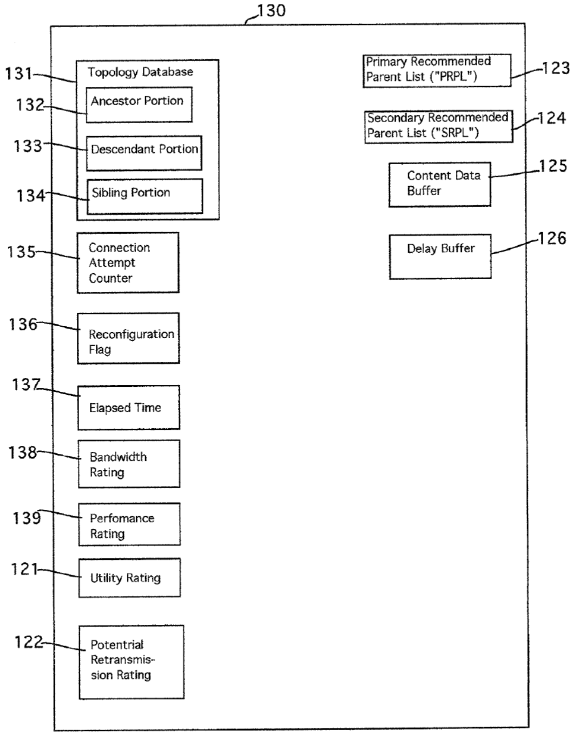


Fig. 13

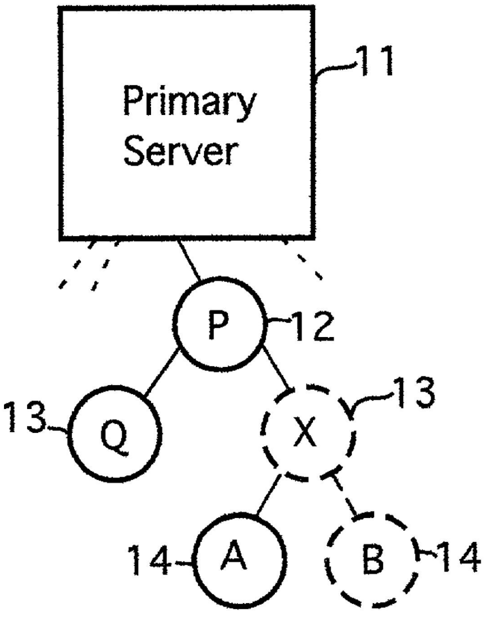


Fig. 23

EX1006, Figures 13, 23.

230. O'Neal further teaches that when a node becomes unavailable, its child nodes can continue communication by interacting with one another rather than relying on the unavailable parent. EX1006, [0168] ("The recipients of the propagation signal … allow its sibling to dock with it as a child node…."). Referring to Figure 23, "[i]n the event that node X were to leave the distribution network, nodes A and B would of course stop receiving content data from node X." EX1006,

[0166]. This event triggers what O'Neal describes as a "reconfiguration event." EX1006, [0166].

231. To reconfigure the network in the event that node X leaves, O'Neal explains that its child node (e.g., node A) "dock[s] (or remained docked), for purposes of receiving content data, with the node" that relays the "reconfiguration event" signal, and then "allow[s] its sibling [e.g., node B] to dock with it as a child node for the purpose of transmitting content data to that child node." EX1006, [0168]. O'Neal provides a concrete example in Figure 24: "[N]ode A … docks with node P" to receive data, while "[n]ode B … docks with node A to receive content data." EX1006, [0170]. Such a "reconfiguration event" is a process by which the topology of the distribution network adapts when a parent node becomes unavailable.

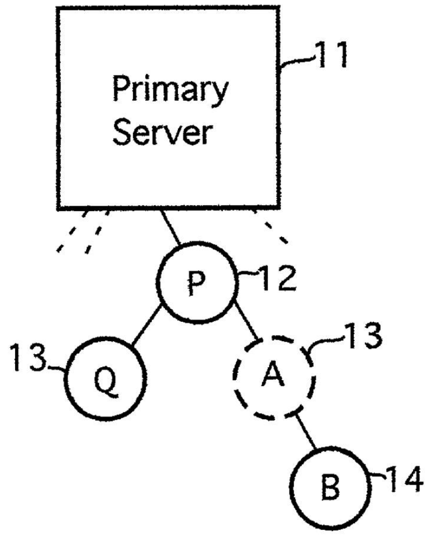


Fig. 24

EX1006, Figure 24.

2. Ogier-Shapiro-Herzog(-Inouchi)[2]-O'Neal Combination

232. As explained in Section VII.C.2 [O'Neal], when a node becomes unavailable, its child nodes stop receiving network content. EX1006, [0166]. Such failures interrupt the flow of data along the distribution tree, leaving the child nodes disconnected, and creating downtime while nodes reconfigure their connections.

233. To maintain communication in that scenario, a POSITA would have recognized the benefit of enabling sibling nodes—child nodes that share the same

---

[2] I placed the parenthetical around Inouchi to indicate that Inouchi is a reference used only when the argument builds on Ground II.

parent—to be aware of one another and to communicate directly. By doing so, one sibling can re-establish upstream connectivity while the other siblings communicates with it to continue receiving data, thereby preserving network service. O'Neal discloses such a reconfiguration process. *See* EX1006, [0166] ("A child node stores information about its sibling (or siblings) in the sibling portion (or sibling database)"); [0168](" The recipients of the propagation signal … allow its sibling to dock with it as a child node…."), [0170] ("Node B changes its propagation rating from red to the next higher propagation rating … and docks with node A to receive content data."); Section VII.C.2 [O'Neal].

234. O'Neal's sibling-aware reconfiguration mechanism improves continuity in a mesh network, for example, the loss of a parent node does not sever communication.

235. It would have been obvious to implement O'Neal's teaching in the Ogier–Shapiro–Herzog(–Inouchi) system so that, when a router becomes unavailable, its child nodes can communicate directly. Ogier already teaches that each node *i* maintains, "[f]or each node u other than node i," "[a] list of children nodes of node i." EX1003, [0065], [0085]; *see also* Sections VII.A.1 and VII.A.5.i)[Ogier and 9H]. A POSITA would have understood that Ogier's maintenance of child-node lists also comports with the basic framework of parent–

child relationships. Incorporating O'Neal's mechanism into this framework would simply extend Ogier's data structures to include awareness of sibling relationships.

236. O'Neal teaches that "a parent node reports to each of its child nodes the addresses of their siblings" (EX1006, [0120]) and that each "child node stores information about its sibling (or siblings) in the sibling portion (or sibling database) 134." EX1006, [0165]. A POSITA would have recognized that by combining this teaching with Ogier's existing list of children, routers in the Ogier–Shapiro–Herzog–Inouchi system would easily be modified to share sibling information. This would have required no change to Ogier's routing algorithms, but only the addition of sibling data to the information already maintained and exchanged among nodes. The predictable result is that when a parent router becomes unavailable, its child nodes would be able to identify their siblings and establish direct communication, thereby maintaining continuity of service.

237. For example, when node *i* in the example discussed above becomes unavailable, one of its child nodes (e.g., node *j*) can communicate directly with its sibling nodes by allowing them to "dock with" node *j*. *See* EX1006, [0168], [0170]. This localized failover mechanism allows the child nodes to remain connected to the subnet and continue receiving data through node *j*.

a) Reasons To Combine Ogier-Shapiro-Herzog(-Inouchi) with O'Neal

238. A POSITA would have been motivated to implement O'Neal's localized failover mechanism in the Ogier-Shapiro-Herzog(-Inouchi) system as described.

239. ***First***, a POSITA would have understood that O'Neal's localized reconfiguration reduces delay in responding to link failures. In Ogier–Shapiro–Herzog, when a node's parent becomes unavailable, the process requires: (1) disseminating the failed parent node's link-state information to all other nodes in the network (EX1003, Abstract, [0010], [0047]); and (2) recomputing the routing path and path tree in order to select a new parent node EX1003, [0197]–[0198]. Although this global recomputation is effective, it introduces latency before more speedy data flow can resume.

240. By contrast, O'Neal's sibling-docking method provides a localized response, allowing data to continue flowing without waiting for a network-wide update. EX1006, [0168], [0170]. This local mechanism provides continuity of communication during the interval before Ogier's global link-state dissemination and path recomputation are completed, resulting in less interruptions in communication.

241. Accordingly, a POSITA would have been motivated to integrate O'Neal's failover process into the Ogier–Shapiro–Herzog network to quickly address local parent-node failures, thereby reducing communication interruptions.

242. ***Second***, a POSITA would have understood that the Ogier–Shapiro–Herzog(–Inouchi) system and O'Neal are compatible. Ogier already provides that each node maintain topology information, including a list of its children nodes. EX1003, [0065], [0085]. O'Neal's approach simply extends this framework by requiring the parent node to inform its children of the addresses of their siblings, which the children then store in a "sibling database." EX1006, [0165]. A POSITA would have recognized that this extension fits naturally within Ogier's existing data structures and messaging framework, requiring only minor modifications to the information distributed by the parent.

243. Because the underlying data list is already in place in Ogier, incorporating O'Neal's sibling-awareness mechanism would not have required redesigning Ogier's path-selection algorithms or topology dissemination process. Instead, it would have employed the same parent–child relationships that Ogier maintains, with the parent providing additional sibling information. A POSITA would have appreciated that this compatibility reduces implementation complexity

and cost, while providing the clear benefit of localized resilience to parent-node failures.

244. A POSITA would have had a reasonable expectation of success in incorporating O'Neal's localized failover mechanism into Ogier-Shapiro-Herzog(-Inouchi).

245. ***First***, as explained above, Ogier–Shapiro–Herzog–(Inouchi) and O'Neal are compatible systems. A POSITA would have found it straightforward to apply O'Neal's technique within Ogier–Shapiro–Herzog–Inouchi so that: (1) the parent node shares its list of child nodes (as taught by Ogier, EX1003, [0065]) with its child nodes (as taught by O'Neal, EX1006, [0120]); and (2) the child nodes communicate directly with one another when the parent becomes unavailable (as taught by O'Neal, EX1006, [0168], [0170]).

246. A POSITA would have recognized that this combination merely integrates O'Neal's sibling-awareness mechanism into Ogier's existing parent–child structure, requiring only minor modifications to information that is already exchanged. When a parent node fails, the child nodes can temporarily maintain communication directly to one another, thereby sustaining data flow until Ogier's global link-state update and path recomputation occur. This outcome is what a

POSITA would have expected—improved resilience without altering the fundamental operation of Ogier's shortest-path routing framework.

247. ***Second***, a POSITA would have recognized that the combination of Ogier–Shapiro–Herzog–Inouchi and O'Neal merely amounts to applying a known technique (O'Neal's localized failover method) to improve a similar system (the subnet routing structure of Ogier–Shapiro–Herzog–Inouchi) in the same way (by providing a communication link between sibling nodes when their parent node becomes unavailable).

248. A POSITA would have understood that integrating O'Neal's technique provides a straightforward, well-known enhancement—localized failover—within an otherwise similar parent–child routing hierarchy. The predictable result is that the subnet responds more quickly to local failures without disrupting Ogier's global path-selection process.

3. Claim 10

a) [10] The wireless mesh network of claim 9 wherein the dataset of child node identifiers is accessible to each child node.

249. Ogier-Shapiro-Herzog(-Inouchi)-O'Neal teaches the additional limitation of claim 10.

250. As explained in Section VII.A.5.i) [9H], Ogier teaches that each node *i* (e.g., routing node 14) maintains, "[f]or each node u other than node i," "[a] list of children nodes of node i, denoted children_i(u)." EX1003, [0065], [0085]. Building on this structure, and as explained in Section VII.C.2 [Ogier–Shapiro–Herzog(–Inouchi)–O'Neal], O'Neal teaches that node *i* "report[s] to each of its child nodes the addresses of their siblings" (EX1006, [0120]) and that each "child node stores information about its sibling (or siblings) in the sibling portion (or sibling database) 134 of its topology database." EX1006, [0165]. O'Neal provides a concrete example: "In the example shown in FIG. 23, nodes Q and X know that they are each other's siblings and nodes A and B know that they are each other's siblings." EX1006, [0165].

.

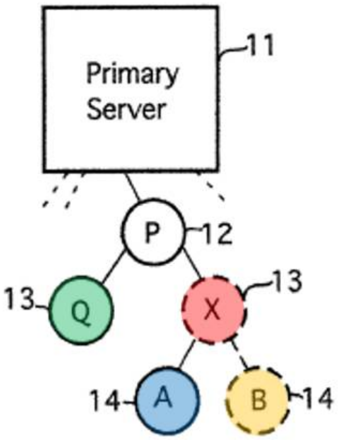


Fig. 23

EX1006, Figure 23.

1. Claim 11

a) [11] The wireless mesh network of claim 10 wherein the AP node dataset of child nodes contains two or more child nodes wherein the zero or more clients in communication with a first child node sends data wherein a destination of the data is a second child node, the first child node sends the data directly to the second child node.

251. Ogier-Shapiro-Herzog(-Inouchi)-O'Neal teaches the additional limitation of claim 11.

252. ***First***, Ogier-Shapiro-Herzog(-Inouchi)-O'Neal teaches "*wherein the AP node dataset of child nodes contains two or more child nodes*." As explained in Section VII.A.5.i) [9H], Ogier teaches that each node *i* maintains, "[f]or each node u other than node i," "[a] ***list of children nodes*** of node i, denoted children_i(u)." EX1003, [0065], [0085]. Ogier further establishes that when link states are updated, "each node 18 in the subnet 10 computes a parent node and ***children*** nodes...," thereby teaching that a given parent may have multiple children. EX1003, [0059]. Indeed, in Ogier-Shapiro-Herzog(-Inouchi)-O'Neal, node *i* would "report[] to ***each of its child nodes*** the addresses of their siblings." EX1006, [0120]; *see also* Figure 23.

253. ***Second***, Ogier teaches that "*the zero or more clients in communication with a first child node sends data wherein a destination of the data is a second child node*." Each node in a subnet—including the "child nodes" discussed above—communicates both with IP hosts 12 (EX1003, [0046], [0050]) and with other nodes in the subnet. Ogier also explains that "[i]n the subnet 10, each node 18 can establish connectivity with one or more other nodes 18." EX1003, [0043].

254. Accordingly, in Ogier's system, the "*destination of the data*" received by the "*first child node*" from the "*zero or more*" clients would be the "*second child node*."

255. ***Third,*** Ogier-Shapiro-Herzog(-Inouchi)-O'Neal teaches that "*the first child node sends the data directly to the second child node*." As explained in Section VII.C.2 [Ogier–Shapiro–Herzog–Inouchi–O'Neal], when the parent node is unavailable, one child node ("*first child node*") sends data directly to its siblings ("*second child node*") by having them "dock with" the first. *See* EX1006, [0168], [0170] ("Node B ["*second child node*"] … docks with node A ["*first child node*"] to ***receive content data***.").

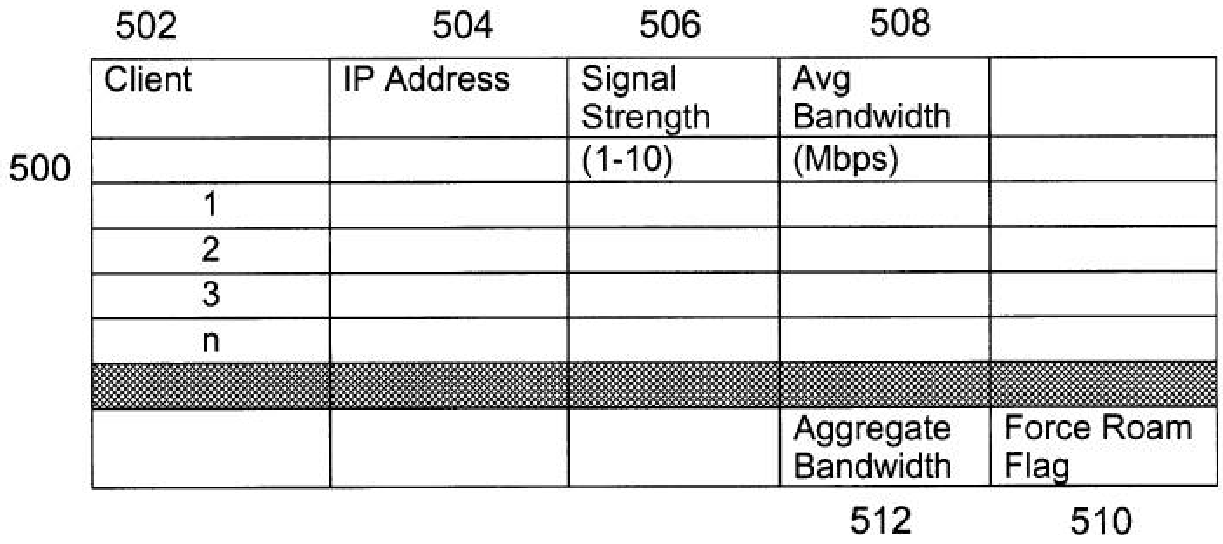
D. Ground IV: Ogier In View Of Shapiro, Herzog, And Cromer ("Ogier-Shapiro-Herzog-Cromer") Renders Obvious Claims 1–7.

1. Cromer (EX1007)

256. Cromer teaches an apparatus and method for "dynamic load balancing of network bandwidth between access points in an 802.11 wireless LAN." EX1007, Abstract. In particular, Cromer teaches a load-balancing technique in which an Access Point "transfer[s] the [client's] connection to the less-congested Access Point thus forcing the client to roam." EX1007, [0030].

257. To enable dynamic load balancing, Cromer teaches that each Access Point maintains a table "storing information relative to each client device that is connected to the Access Point." EX1007, [0038], Fig. 5. Column 508 specifically records the "average bandwidth used by each client," thereby giving the AP node visibility into the traffic demand imposed by each associated device. In addition, entry 512 records the "aggregate bandwidth which is the sum of all the bandwidth used by each client." EX1007, [0038].

Fig 5



Client	IP Address	Signal Strength	Avg Bandwidth	
		(1-10)	(Mbps)	
1				
2				
3				
n				
▨	▨	▨	▨	▨
			Aggregate Bandwidth	Force Roam Flag

EX1007, Figure 5.

258. An Access Point can use the table 500 to perform dynamic load balancing. Cromer explains that if the total client demand on an AP, reflected in the aggregate bandwidth stored at entry 512, "exceeds a predetermined threshold," that AP initiates a request to offload one or more of its clients to another AP. EX1007, [0038] ("the force roam flag location which is set when the aggregate bandwidth in 512 exceeds a predetermined threshold. It is the setting of this flag that causes an access point to interrogate adjoining access points to see if it can offload client devices to other access points.."); [0041] ("the aggregate bandwidth is compared against a threshold value. If the aggregate bandwidth is equal or above the threshold the program enters block 612 where it sets the redistribution flag"); [0048] ("If the

access point is not available the program loops. If access point is available the program exits … whereat the client establishes a link with the access point to which it is forced.”). The receiving AP then evaluates whether it can accommodate a new client by calculating its “extra capacity,” which is the “difference between its Max. Capacity and the aggre[g]ate capacity 512” at the receiving AP. EX1007, [0050]. If extra capacity exists, the receiving AP accepts the client. EX1007, [0050] (“If extra capacity remains the device is accepted.”).

2. Ogier-Shapiro-Herzog-Cromer Combination

259. The Ogier–Shapiro–Herzog–Cromer combination builds on the Ogier–Shapiro–Herzog framework described in Section VII.A.4 [Ogier–Shapiro–Herzog]. In that system, Ogier’s routers determine path trees and routing paths by “running a shortest-path algorithm such as Dijkstra’s algorithm” using the topology table TT_i, which stores link-state information including a cost between two neighboring nodes. EX1003, [0058], [0061], [0198]. As explained in Section VII.A.4, this link cost is computed as a weighted sum of: (1) “one, for minimum-hop routing, or the link delay plus a constant bias” (EX1003, [0044]); and (2) the goodness factor (GF) as taught by Shapiro (EX1004, [0048]).

260. Shapiro further explains that “[t]he goodness factor represents any number of qualitative and quantitative feature of the corresponding route” and can

reflect "the status of the communications path between a series of nodes." EX1004, [0048].

261. It was well known in the art that the "status of the communications path between a series of nodes" depends in part on node capacity. Cromer teaches that each access point (AP) has a maximum data traffic capacity, i.e., "Max. capacity." EX1007, [0050]. When an AP reaches a threshold amount of data traffic, it would offload one or more clients so that it retains sufficient "bandwidth to service the [remaining] clients." EX1007, [0014]. Cromer further explains that before accepting an additional client, the AP first determines whether it has "extra capacity" and will accept the new client only if that extra capacity exists. EX1007, [0050].

262. It would have been obvious to combine the Ogier-Shapiro-Herzog system with Cromer so that Shapiro's goodness factor favors links to nodes with greater extra capacity and excludes nodes exceeding their maximum bandwidth, (e.g., setting the cost of the link as "infinite" as taught by Ogier (EX1003, [0101])).

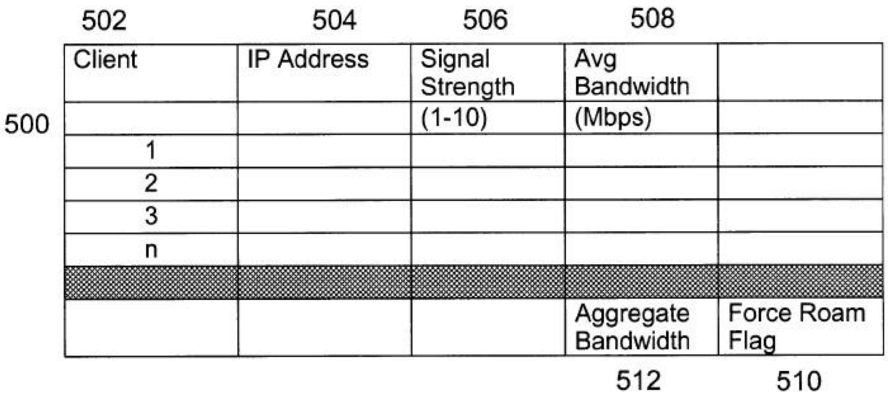
263. For example, each router in the Ogier-Shaprio-Herzog-Cromer subnet (e.g., router *u*, as taught by Ogier) would maintain a table like Cromer's Table 500, including the "aggregate bandwidth" entry 512. EX1007, [0038]. Using this information, router *u* can compute its "extra capacity" ("EC"), defined as the difference between its maximum bandwidth and its current bandwidth usage.

EX1007, [0050] ("extra capacity is determined by difference between the Max. Capacity and the aggre[g]ate capacity 512.").

264. The routing node *u* would then distribute its computed EC to all other nodes in the subnet 10 using Ogier's TBRPF protocol, which disseminates link-state information throughout the network. EX1003, [0047] ("Each router 14 in the subnet 10 is responsible for detecting, updating, and reporting changes in cost and up-or-down status of each outgoing communication link to neighbor nodes."). When another node *i* receives node *u*'s EC value, node *i* updates the link costs for all links connected to node *u* to reflect the new EC. A POSITA would have recognized that incorporating EC in this manner integrates congestion awareness into the combination's routing metrics.

265. The predictable result is that the route-selection process favors paths with higher EC (and therefore less congestion and greater available bandwidth) and excludes routes through nodes with negative EC (by making the link cost "infinite" and "deleting" it from the topology table, as discussed in Ogier (EX1003[0101])).

Fig 5



Client	IP Address	Signal Strength	Avg Bandwidth	
		(1-10)	(Mbps)	
1				
2				
3				
n				
▨	▨	▨	▨	▨
			Aggregate Bandwidth	Force Roam Flag

EX1007, Figure 5.

a. Reasons To Combine Ogier-Shapiro-Herzog With Cromer

266. A POSITA would have been motivated to combine the Ogier-Shapiro-Herzog(-Inouchi) system with Cromer for four reasons.

267. ***First***, as explained above, Shapiro teaches that the goodness factor can reflect "the status of the communications path between a series of nodes." EX1004, [0048]. A POSITA would have understood that one such status of a communications path is a node's extra capacity (EC)—its available bandwidth to service additional traffic. Cromer articulates this point by teaching that nodes should not be overloaded beyond their maximum capacity and should offload clients when bandwidth reaches a threshold. *See* EX1007, [0038] ("the force roam flag location which is set when the aggregate bandwidth in 512 exceeds a predetermined threshold. It is the setting of this flag that causes an access point to interrogate adjoining access points to see

if it can offload client devices to other access points..”); [0041] (“the aggregate bandwidth is compared against a threshold value. If the aggregate bandwidth is equal or above the threshold the program enters block 612 where it sets the redistribution flag”); [0048] (“If the access point is not available the program loops. If access point is available the program exits … whereat the client establishes a link with the access point to which it is forced.”).

268. Accordingly, a POSITA would have been motivated to include EC into Shapiro’s goodness factor so that the computed route path and path tree reflect each node’s ability to carry traffic, favoring paths that traverse nodes with higher EC and excluding paths that include nodes whose bandwidth usage exceeds the maximum capacity. By doing so, the routing algorithm would dynamically adjust to congestion and resource availability in the network. To implement this, a POSITA would have been motivated to include Cromer’s capacity table in each of Ogier’s nodes and to incorporate those EC values into the cost metrics disseminated and used in Ogier’s TBRPF protocol, as described above.

269. ***Second***, Cromer provides motivation for the combination. Cromer discloses that each node has a maximum amount of traffic it is capable of servicing, and when a node operates at or near that capacity, it is instructed to offload one or more clients such that it retains sufficient “bandwidth to service the [remaining]

clients." EX1007, [0014]. A POSITA would have understood that congestion affect the quality of service on a communications path. Accordingly, a POSITA would have been motivated to modify Shapiro's goodness factor so that path selection accounts for available extra capacity, preferring paths through nodes with sufficient EC and avoiding paths through nodes operating at or near their maximum capacity.

270. ***Third***, combining Ogier–Shapiro–Herzog with Cromer merely amounts to applying a known technique to improve a similar system. Shapiro teaches modifying the cost metric to "account [for] the quality or speed of a link" (EX1004, [0008]), and a POSITA would have recognized that one way to implement this teaching is by incorporating traffic capacity—specifically, Cromer's extra capacity (EC)—into the link-cost calculation. Doing so would have improved Ogier–Shapiro–Herzog's routers, which already compute paths based on link costs and were thus ready for such improvement. This would have predictably resulted in route selection that would prefer paths traversing nodes with greater extra capacity, while excluding paths through nodes with load above the maximum capacity. A POSITA would have known that the combination would have yielded predictable results (selecting a routing path or path tree based also on extra capacity EC of nodes along the path).

271. ***Fourth***, combining Ogier–Shapiro–Herzog with Cromer merely amounts to a simple substitution of one known element for another. Substituting a goodness factor that accounts for both throughput and EC in place of a goodness factor that accounts only for throughput would have been a straightforward modification. The predictable result is that the routing path or path tree is selected based on congestion and throughput, rather than throughput alone. This substitution would have yielded the expected improvement of excluding overloaded nodes while favoring high-throughput paths and paths with nodes carrying more extra capacity, without requiring any significant change to Ogier's underlying routing framework.

272. A POSITA would have had a reasonable expectation of success in combining Cromer with the Ogier-Shapiro-Herzog system in the above manner.

273. ***First***, just as the combination of Ogier and Shapiro (see Section VII.A.4.a) [Ogier–Shapiro]), combining Ogier–Shapiro–Herzog with Cromer merely amounts to modifying the calculations used to determine link costs that are input to Ogier's route-selection process and path-tree generation. A POSITA would have recognized that such a modification predictably results in selection of routing paths that optimize for the modified costs.

274. Ogier teaches that paths are selected by applying Dijkstra's algorithm: "One exemplary path selection algorithm is to apply Dijkstra's algorithm to compute

shortest paths (with respect to cost, c) to all destinations." EX1003, [0198]; *see also* [0084] ("Node i then computes (step 104) the parent nodes p i(u) for all potential source nodes src by running a shortest-path algorithm such as Dijkstra's algorithm."). Dijkstra's algorithm is agnostic to the definition of the cost metric, and that optimization using different cost components—such as incorporating Cromer's extra capacity (EC)—would be straightforward.

275. Indeed, it was well known in the art that Dijkstra's algorithm reliably finds optimal-cost routes under whatever cost definition is provided. EX1013, 13:51–67 ("Typically, an algorithm is used to generate the overall routing table of FIG. 5. One such algorithm is the Dijkstra algorithm…. Using the algorithm, finding the shortest (i.e.,the route which minimi[z]es a metric) route for travelling from a given vertex on a graph to every other vertex is possible.… In the case where the metric represents the cost of each link, the algorithm returns an indication of the least cost route from the root vertex to the particular vertex as well as the overall cost of the route.").

276. ***Second***, both Ogier and Shapiro teach that path selection can incorporate different types of information into the link-cost calculation. Ogier explains that "***[a]ny technique*** for assigning costs to links can be used to practice the invention." EX1003, [0044]. Shapiro likewise teaches that the goodness factor

"represents any number of qualitative and quantitative feature of the corresponding route." EX1004, [0048].

277. Accordingly, a POSITA would have had a reasonable expectation that modifying the link-cost computation in the Ogier–Shapiro–Herzog system to also account for Cromer's extra capacity (EC) would be effective in selecting preferred paths. Because Dijkstra's algorithm optimizes over whatever cost metric is provided, incorporating EC into the link costs would predictably yield paths that both balance traffic and avoid congested nodes, fully consistent with the flexible cost framework already contemplated by Ogier and Shapiro.

278. ***Third***, a POSITA would have understood that Ogier's path-selection technique is carried out with program code as it relies on running Dijkstra's algorithm. Given this, a POSITA would have recognized that the Ogier–Shapiro–Herzog–Cromer combination would have merely required writing software instructions to adjust the link-cost calculation—by incorporating Cromer's extra capacity (EC) alongside hop count, delay, and throughput—before executing the Dijkstra algorithm. Such a modification does not alter the underlying algorithm or routing process; it only changes the cost values that are input. A POSITA would have considered this a straightforward programming task well within the level of ordinary skill.

3. Claim 1

279. Claim 1 is identical to claim 9 except for limitation [1H]. Ogier-Shapiro-Herzog renders obvious claim limitations [1Pre]-[1G] for the reasons discussed above for claim limitations [9Pre]-[9G], respectively.

a) [1H] wherein one of the datasets contained in the AP node comprises an amount of AP data traffic wherein the amount of AP data traffic comprises the amount of data exchanged between the AP node and the zero or more clients in communication with the node and wherein another of the datasets contained in the AP node comprises a maximum capacity amount of the AP node.

280. Ogier-Shapiro-Herzog-Cromer teaches [1H].

281. ***First***, Ogier-Shapiro-Herzog-Cromer teaches "*wherein one of the datasets contained in the AP node comprises an amount of AP data traffic wherein the amount of AP data traffic comprises the amount of data exchanged between the AP node and the zero or more clients in communication with the node.*"

282. For example, Cromer discloses that each router includes table 500 with entry 512, which "stores the ***aggregate bandwidth*** which is the sum of all the bandwidth ***used by each client***." EX1007, [0038]. This means that each router maintains a dataset tracking how much traffic is being used by its connected clients. Accordingly, Cromer's table 500 constitutes a "*dataset*," and entry 512 corresponds

to a dataset element comprising "*the amount of data exchanged between the AP node and the zero or more clients in communication with the node*."

283. ***Second***, Ogier-Shapiro-Herzog-Cromer teaches "*wherein another of the datasets contained in the AP node comprises a maximum capacity amount of the AP node*." As explained in Section VII.D [Ogier–Shapiro–Herzog–Cromer], each router computes its extra capacity (EC) and distributes that value to every other node in subnet 10. To calculate EC, Cromer teaches that a router determines the "difference between the Max. Capacity and the aggre[g]ate capacity 512." EX1007, [0050] ("[E]xtra capacity [of an Access Point] is determined by difference between the Max. Capacity and the aggre[g]ate capacity 512."). Thus, to compute its EC, each router maintains a "Max. Capacity" value, which corresponds to "*a maximum capacity amount of the AP node*."

4. Claim 2

a) [2] The wireless mesh network of claim 1 wherein one of the datasets contained in the AP node comprises a cost to connect and said cost to connect is calculated using the level of network congestion experienced by the available parent node.

284. Ogier-Shapiro-Herzog-Cromer teaches the additional limitation of claim 2.

285. As explained in Section IX.A.1 [Ogier], each router maintains "[a] topology table, denoted TT_i" that records, for each link between nodes *u* and *v*, "the cost associated with the link." EX1003, [0060]–[0061]. Therefore, TT_i stores the link cost between the router itself (node *i*) and each of its available parent nodes, and that these costs are used to compute shortest paths or constructing the path tree. EX1003, [0084] ("[N]ode i enters (step 100) the new link-state information, if any into the topology table…Node i then computes (step 104) the parent nodes p i(u) for all potential source nodes src by running a shortest-path algorithm such as Dijkstra's algorithm.").

286. Furthermore, as described in Section VII.D.2 [Ogier–Shapiro–Herzog–Cromer], a POSITA would have found it obvious to revise these link costs to incorporate additional metrics such as Cromer's extra capacity (EC) of the node to which the link is connected (i.e., "*available parent nodes*"). See Section VII.D.2 [Ogier–Shapiro–Herzog–Cromer].

5. Claim 3

a) [3] The wireless mesh network of claim 1 wherein the communication criteria comprises instructions for the AP node to select the associated parent node wherein the cost to connect of the associated parent is the lowest of all available parent nodes.

287. Ogier-Shapiro-Herzog-Cromer teaches the additional limitation of claim 3.

288. ***First***, Ogier teaches "*wherein the communication criteria comprises instructions for the AP node to select the associated parent node*" for the reasons provided in Section VII.A.5.d) and VII.A.5.f) [9C and 9E]. For example, a router (such as node F in Figure 1) choose its associated parent node (e.g., node H) by minimizing the total link cost of all identified neighboring node that lies along the path toward the destination (gateway 16). *See* EX1003, [0058] ("embodiments of the TBRPF protocol can use … ***shortest path trees***"); [0084] ("Node i then computes (step 104) the parent nodes p i(u) for all potential source nodes src by running a ***shortest-path algorithm*** such as ***Dijkstra's algorithm***"); [0197]–[0198] ("One exemplary ***path selection algorithm*** is to apply ***Dijkstra's algorithm*** to compute shortest paths (with respect to cost, c) to all destinations.").

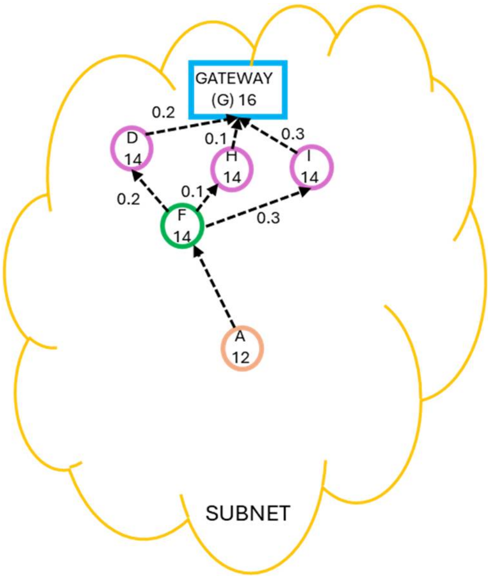
289. ***Second***, Ogier-Shapiro-Herzog-Cromer also teaches that the "*associated parent node*" is selected "wherein *the cost to connect of the associated*

parent is the lowest of all available parent nodes." For example, Ogier teaches that a router can choose a path to a destination node using a "path selection algorithm" that "appl[ies] Dijkstra's algorithm to compute shortest path[] (with respect to cost, c) to [the destination node]." EX1003, [0198]. Similarly, Ogier discloses that a routing node can update the "shortest path tree" for TBRPF (EX1003, [0058]) by "run[ning] a shortest-path algorithm such as Dijkstra's algorithm" (EX1003, [0084]) on the topology table TT_i.

290. As discussed above, the topology table provides the link-state information, including the costs of links to neighboring nodes, and that Dijkstra's algorithm is applied over those costs to compute the shortest paths. Because Dijkstra's algorithm identifies the minimum-cost path to each destination, a POSITA would have recognized that the parent node chosen for forwarding is most likely the neighbor whose link cost is "*the lowest of all available parent nodes.*"

291. At a minimum, for certain network configurations, Dijkstra's algorithm would choose as the parent node the neighbor whose link cost is "*the lowest of all available parent nodes*." For example, when Ogier's subnet 10 is arranged as illustrated below, with link costs depicted as edge weights, the router at node F computes shortest paths to the gateway G. In this arrangement, the total link cost along the path F–H–G (=0.2) is lower than the total link costs along paths F–D–G

(=0.4) and F–I–G (=0.6). Applying Dijkstra's algorithm, node F therefore selects node H as its parent, because the route through H provides the minimum-cost path to the destination. And in this case, the link cost to the associated parent node H is "*the lowest of all available parent nodes*" D, H, and I.



EX1003, Figure 1.

6. Claim 4

a) [4] The wireless mesh network of claim 1 wherein the communication criteria comprises instructions for the AP node to associate with an available parent node wherein the amount of AP data traffic is less than the sum of the maximum capacity amount of the available parent node and the available parent amount of AP data traffic.

292. During prosecution, Applicant introduced the additional limitation of claim 4 as claim 8 of the Preliminary Amendment filed on December 8, 2009.

EX1008, 115. In reference to the newly introduced amendment, Applicant stated that "[i]n regards to claims 7 and 8, ¶45 teaches that each node, prior to selecting a parent for association, checks the capacity of the available parent as well as current network congestion levels of the available parent, and ***only associates with an available parent whose capacity has not been exceeded***." EX1008, 118.

293. If this limitation of claim 4 simply requires "check[ing] the capacity of the available parent as well as current network congestion levels of the available parent, and only associat[ing] with an available parent whose capacity has not been exceeded," Ogier-Shapiro-Herzog-Cromer also teaches this limitation.

294. ***First***, Ogier teaches that "*the communication criteria comprises instructions for the AP node to associate with an available parent node*" for the reasons provided in Section VII.A.5.d) and VII.A.5.f) [9C and 9E]. For example, a router (e.g., node F in Figure 1) selects its associated parent node (e.g., node H) by identifying the neighboring node that lies along the path toward the destination (gateway 16) and that minimizes the total link cost. *See* EX1003, [0058] ("embodiments of the TBRPF protocol can use … ***shortest path trees***"); [0084] ("Node i then computes (step 104) the parent nodes p i(u) for all potential source nodes src by running a ***shortest-path algorithm*** such as ***Dijkstra's algorithm***");

[0197]–[0198] ("One exemplary ***path selection algorithm*** is to apply ***Dijkstra's algorithm*** to compute shortest paths (with respect to cost, c) to all destinations.").

295. ***Second***, Ogier-Shapiro-Herzog-Cromer also teaches that the "*available parent node*" is associated with the AP node "*wherein the amount of AP data traffic is less than the sum of the maximum capacity amount of the available parent node and the available parent amount of AP data traffic*." For example, in Figure 1 of Ogier, node F has no child nodes connected. As a result, the "*amount of AP data traffic*" associated with node F is zero. Because this value at node F is zero, it is less than the "*sum of the maximum capacity amount of the available parent node [node D, H, or I] and the available parent amount of AP data traffic*" because the maximum capacity amount for the available parent node (D, H, or I) would be non-zero.

296. Furthermore, a POSITA would have understood that a router in the combined system would only associate with an available parent whose capacity has not been exceeded. As explained in Section VII.D.2 [Ogier–Shapiro–Herzog–Cromer], when an available parent's capacity is exceeded, the system assigns the link cost between the router and that parent as "infinite." EX1003, [0101]. In a shortest-path algorithm such as Dijkstra's, links with infinite cost are excluded from consideration, meaning that any path (and thus any potential parent node) associated

with an infinite link cost cannot be selected. EX1003, [0101] ("failed links (represented by an infinite cost) and links that are unreachable are deleted from the topology table TT_i.").

7. Claim 5

a) [5] The wireless mesh network of claim 4 wherein the communication criteria comprises instructions for the AP node to associate with a single suitable parent node wherein a parent node is suitable if the throughput capacity of the parent node is the highest of all available parent nodes.

297. During prosecution, Applicant introduced the additional limitation of claim 5 as claim 11 of the Preliminary Amendment filed on December 8, 2009. EX1008, 116. In reference to the newly introduced amendment, Applicant stated that, "[c]laims 10 and 11 relate to throughput capacities. … Maximizing throughput as the parent selection criterion is discussed in ¶48." EX1008, 119. Paragraph 48 of the originally submitted specification states: "To answer this, consider the network in Fig. 4B. If the parameter set by the Access Server is to maximize throughput. Node 002 would examine all nodes it can connect to and choose a parent that ensures the ***highest global throughput*** (GT)." EX1008, 14.

298. If the additional limitation of claim 5 requires "choos[ing] a parent that ensures the highest global throughput (GT)," Ogier-Shapiro-Herzog-Cromer teaches it.

299. ***First***, Ogier teaches "*wherein the communication criteria comprises instructions for the AP node to associate with a single suitable parent node*" for the reasons discussed in Section VII.A.6.a) [12H].

300. ***Second,*** Ogier-Shapiro-Herzog-Cromer also teaches "*wherein a parent node is suitable if the throughput capacity of the parent node is the highest of all available parent nodes*." As explained in Section VII.A.4.b)a [Ogier–Shapiro–Herzog], the goodness factor may be defined as "a decaying average of periodically sampled ***throughput*** for a node." EX1004, [0048]. Accordingly, when Herzog's policy server defines the goodness factor in this way and sets the weight W to a sufficiently high value, Dijkstra's algorithm (EX1003, [0084], [0198]) would find a path that maximizes the throughput along the path, i.e., global throughput.

8. Claim 6

a) The wireless mesh network of claim 1 wherein one of the datasets contained in an AP node comprises a throughput capacity of the AP node.

301. During prosecution, Applicant introduced the additional limitation of claim 6 as claim 10 of the Preliminary Amendment filed on December 8, 2009. EX1008, 115. In reference to the newly introduced amendment, Applicant stated that, "[c]laims 10 and 11 relate to throughput capacities. The ***calculation of the throughput capacity at each node*** is discussed in ¶45 which provides a sample

calculation of the throughput at a sample node. (See 'GT Global Throughput: This is the product of all throughputs each node along the route provides. Nodes connected to the root have a throughput related to LS. Thus the throughput of Node 002 in Fig. 4B is 79%.')." EX1008, 119.

302. If the "calculation of the throughput capacity at each node" provides written description support for the additional limitation of claim 6, Ogier-Shapiro-Herzog-Cromer teaches it.

303. As described in Section VII.D.7 [Claim 5], when Herzog's policy server defines the goodness factor as "a decaying average of periodically sampled ***throughput*** for a node" and sets the weight W high enough, Ogier's nodes would use, as their stored link costs, the throughput of the nodes forming the links. Then, when determining routing paths, Ogier's Dijkstra algorithm (EX1003, [0084], [0198]) would calculate a path that maximizes the total throughput of nodes on the path, in other words, ***the global throughput***.

304. Accordingly, the Ogier–Shapiro–Herzog–Cromer combination teaches computing and storing the throughput capacity at each node, and calculating the global throughput across the network.

9. Claim 7

a) The wireless mesh network of claim 1 wherein the means for switching two-way data communications selects the second associated parent node when the first associated parent node amount of AP data traffic approaches the parent node maximum capacity amount within a congestion value set by the access server.

305. Ogier-Shapiro-Herzog-Cromer teaches the additional limitation of claim 7.

306. ***First***, Ogier teaches "*wherein the means for switching two-way data communications selects the second associated parent node*" for the reasons discussed in Section VII.A.5.f)a [9E-1].

307. ***Second***, Ogier-Shapiro-Herzog-Cromer teaches that the "*second associated parent node*" is selected "*when the first associated parent node amount of AP data traffic approaches the parent node maximum capacity amount*." As described in Section VII.D.2 [Ogier–Shapiro–Herzog–Cromer] and Sections VII.D.4 and VII.D.5[Claims 2 and 3], a router chooses path tree and its routing path by identifying the route with the lowest link cost, where the link cost incorporates extra capacity (EC).

308. EC represents the remaining capacity of a node —that is, the difference between its maximum capacity and its current traffic load—for the same reasons

discussed in VII.D.2. When the EC of the currently associated parent node (the "*first associated parent node*") is low—i.e., when "*the first associated parent node amount of AP data traffic approaches the parent node maximum capacity amount*"—the cost of the link to that parent increases. In that case, Ogier's path-selection process, which applies Dijkstra's algorithm over the link costs, would identify and select another parent node (the "second associated parent node") with greater EC, because that node offers a lower-cost path. *See* EX1003, [0084] ("Node i then computes (step 104) the parent nodes p i(u) for all potential source nodes src by running a ***shortest-path algorithm*** such as ***Dijkstra's algorithm***"); [0197]–[0198] ("One exemplary ***path selection algorithm*** is to apply ***Dijkstra's algorithm*** to compute shortest paths (with respect to cost, c) to all destinations.").

309. ***Third***, Ogier-Shapiro-Herzog-Cromer teaches that the "*maximum capacity amount within a congestion value set by the access server*." As an initial matter, the '243 Patent specification contains no written description—either directly or indirectly—for the limitation that the "*maximum capacity amount*" be set by the "*access server*." Therefore, the Ogier–Shapiro–Herzog–Cromer combination discussed in Section VII.D.2[Ogier–Shapiro–Herzog–Cromer] teaches this feature to the same extent that it is supported by the '243 Patent.

310. Moreover, it would have been obvious to further modify Ogier-Shapiro-Herzog-Cromer such that Herzog's policy server ("*access server*") issues the instructions ("*congestion value*") setting the "*maximum capacity amount*" for the routers.

311. As explained in Section VII.A.4 [Ogier–Shapiro–Herzog], Herzog's policy server "translates the [network] policy into a set of lower-level instructions," which it then transmits to the nodes so that they can implement the network policy. *See* EX1005, [0005] ("Network devices understand the instructions, and the specified goals thus can be accomplished."); [0029] ("The policy system 104 … translates the policy rules into a set of lower-level instructions that network devices understand.").

312. It was well known in the art that the "maximum capacity amount" for routers may be established through a network policy set by a policy server like Herzog's. EX1012, 27:16–18 ("Policies can be specified to control traffic flows in terms of overall bandwidth guarantees, ***bandwidth limits***, …."); 50:17–18 ("Policy Server communicates the bandwidth requirement to the Internet router, which supports this function").

313. Accordingly, it would have been obvious to further modify Ogier-Shapiro-Herzog-Cromer for Herzog's policy server to distribute the additional

"instructions" defining the "*maximum capacity amount*" for the routing nodes to use in the EC calculations. A POSITA would have made such modification at least for the same rationale set forth in Section VII.A.4.b)a [Reasons-To-Combine Ogier-Shapiro-Herzog].

E. Ground V: Ogier In View Of Shapiro, Herzog, Cromer, And Inouchi Renders Obvious Claims 1–7.

314. If Patent Owner argues that Ogier-Shapiro-Herzog-Cromer in Ground IV does not render Claims 1–7 obvious because it does not expressly disclose "*a processor and a storage medium with instructions*" as required under the construction of "*means for switching*" (limitation [1E]), that limitation would have been obvious in view of Inouchi for the reasons provided in Section VII.B [Ground II].

VIII. CONCLUSION

315. I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and that these statements were made with knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code.

Dated: August 27, 2015

Christopher Hansen, Ph.D.