

David L. Mortensen (#8242)
Email: dmortensen@foley.com
Tanner B. Camp (#17805)
Email: tcamp@foley.com
FOLEY & LARDNER LLP
95 S. State Street, Suite 2500
Salt Lake City, UT 84111
Telephone: 801.401.8900

Attorneys for Snap One, LLC

**IN THE UNITED STATES DISTRICT COURT
DISTRICT OF UTAH**

<p>SNAP ONE, LLC, a North Carolina limited liability company,</p> <p style="text-align: center;">Plaintiff,</p> <p style="text-align: center;">v.</p> <p>AVA, INC., a Delaware corporation; JOSH.AI, INC., a Delaware corporation,</p> <p style="text-align: center;">Defendants.</p>	<p>COMPLAINT AND JURY DEMAND</p> <p>Case No. 2:23-cv-522</p>
---	---

Plaintiff Snap One, LLC hereby complains, petitions, and alleges against defendants Josh.ai, Inc. (“Josh.ai”) and AVA, Inc. (“AVA”) (collectively, “Defendants”) as follows:

INTRODUCTION

1. In 2019, Snap One spent more than ten million dollars to acquire a company, NEEO AG (“NEEO”), that had developed an innovative smart home remote (the “Neoo Remote”). As part of the acquisition, several former NEEO employees began working for Snap One to develop an upgraded Neoo Remote that, although nearly identical in external design to the previous version, featured several technical upgrades and software changes to improve the

product’s functionality and usability. Snap One released the upgraded Neeo Remote for use with Snap One’s market-leading Control4 smart home systems in November 2019.

2. Less than two years later, a core group of former NEEO leaders, then employed by Snap One, left Snap One and started a new company called AVA. Leveraging the intellectual property purchased or developed by Snap One, and without Snap One’s consent, AVA quickly developed a smart remote of its own (the “AVA Remote”).



3. As demonstrated by the images above, the AVA Remote was strikingly similar to the Neeo Remote. The uncanny resemblance did not go unnoticed. Both remotes were exclusively sold to professional systems integrators (“Dealers”) within the custom electronic design and installation industry (the “CEDIA Industry”). Those Dealers publicly remarked:

“Looks just like a neo [*sic*] remote,” “[T]his new remote is just a Neeo 2.0,” and “Customers might think the AVA remote is a [Neeo] remote and that is not a good thing.” AVA also sought to profit off of Snap One’s name and reputation by, among other things: (i) improperly using Snap One’s Control4 trademark on its website, in marketing hashtags, and in numerous images of the AVA Remote; (ii) representing that AVA was founded by “Former [Snap One] Executives”; and (iii) advertising the AVA Remote in closed Snap One Dealer groups. This conduct led to even more confusion in the marketplace.

4. In March 2022, Snap One demanded that AVA stop its improper conduct and took steps to prevent the AVA Remote from interoperating with Snap One’s Control4 smart home platform. AVA eventually agreed to stop using Snap One’s Control4 marks, but its misconduct did not stop there. A few months later, AVA approached one of Snap One’s certified product development partners, Josh.ai, about a collaborative partnership.

5. Josh.ai had produced a voice control system that, as set forth below, was licensed to integrate with the Control4 platform. Specifically, in 2017, Josh.ai and Snap One executed a Software Development Kit License Agreement (the “SDK Agreement”) that gave Josh.ai a limited license to use certain Snap One software, code, web services, application programming interfaces (“APIs”), and other development resources (collectively, the “Development Resources”) to build applications for use with its products to integrate with Control4 software. Pursuant to the SDK Agreement, Josh.ai had used the Development Resources to integrate its voice control system with Snap One’s Control4 platform.

6. Knowing this, in the fall of 2022, AVA partnered with Josh.ai to embed Josh.ai's interface within the AVA Remote and integrate the remote with Josh.ai systems, thereby gaining unauthorized access to, and interoperability with, Snap One's Control4 platform.

7. Soon thereafter, on June 27, 2023, Josh.ai announced the release of the "Josh Remote." According to the release, AVA and Josh.ai jointly developed the Josh Remote, which appears to leverage the AVA Remote's hardware, firmware, and industrial design, causing additional market confusion.



8. Both the AVA Remote and the Josh Remote (together, the "Infringing Remotes") improperly incorporate Snap One's valuable intellectual property. Among other things, they infringe on Snap One's patents, incorporate Snap One's valuable trade secrets and confidential proprietary information, and copy the Neeo Remote's unique and innovative design. Beyond

that, both Josh.ai and AVA produce additional products that also infringe on Snap One's valuable patent rights.

9. This is not the only instance of Josh.ai's improper conduct in its relationship with Snap One. In 2021, Josh.ai entered into a Distributor Agreement with Snap One, in which it granted Snap One the exclusive right to sell certain Josh.ai products to Control4 Dealers. However, contrary to its repeated representations to Snap One, Josh.ai failed to refer Control4 Dealers that desired to purchase Josh.ai products to Snap One. Instead, it breached its contractual obligations by actively marketing and selling its products to new and existing Control4 Dealers at prices that were lower than Snap One (as a reseller of the products) could offer. In so doing, Josh.ai improperly took advantage of Snap One's sales and marketing efforts to acquire additional customers and left Snap One with Josh.ai products it could not sell.

10. Defendants' misconduct should be seen for what it is: an effort to exploit Snap One's intellectual property and goodwill and bypass the years of work and effort necessary to develop market-leading smart home products. Snap One gave Defendants multiple chances to correct or stop their behavior and provided notice of their breaches of contract. They disregarded those notices. Consequently, Snap One now has no choice but to seek relief from the legal system to protect its intellectual property and contractual rights.

THE PARTIES, JURISDICTION, AND VENUE

11. Plaintiff Snap One, LLC is a North Carolina limited liability company with its co-headquarters and principal place of business in Lehi, Utah.

12. Defendant AVA, Inc. is a Delaware corporation with its principal place of business in Park City, Utah.

13. Defendant Josh.ai, Inc. is a Delaware corporation with its principal place of business in Denver, Colorado.

14. The Court has subject matter jurisdiction over this matter pursuant to 28 U.S.C. § 1338(a) because it is a civil action arising under an Act of Congress relating to patents. The Court also has subject matter jurisdiction pursuant to 28 U.S.C. § 1331(a) because Defendants' conduct violates at least the Defend Trade Secrets Act ("DTSA") (18 U.S.C. § 1836) and the Lanham Act (15 U.S.C. § 1125(a)). The Court has supplemental jurisdiction over Snap One's state law claims pursuant to 28 U.S.C. § 1367. The state law claims are integrally related to Snap One's federal claims and arise from a common nucleus of operative facts, such that the administration of the state law claims with the federal claims furthers the interest of judicial economy.

15. This Court has personal jurisdiction over Defendants pursuant to Utah Code § 78B-3-205 because Defendants have transacted business within the State of Utah and caused injury within the State of Utah. This dispute arises from Defendants' contacts with this District, and Defendants have purposefully availed themselves of the Utah forum such that they could reasonably anticipate being called to defend themselves in courts within the state. Additionally, Defendants transact substantial business with entities and individuals in Utah by, among other things, committing acts of patent infringement and inducing others to commit acts of patent infringement in Utah, such as by importing, offering to sell, distributing, and selling products that infringe the Snap One Patents, including the Infringing Remotes, in Utah. Thus, requiring Defendants to litigate this case in this Court does not offend traditional notions of fair play and substantial justice and is permitted by the Due Process Clause of the United States Constitution.

16. Venue in this Court is proper under 28 U.S.C. § 1391(b)(2) because a substantial part of the events giving rise to the claims in this Complaint occurred in this District. Venue in this Court is also proper under 28 U.S.C. § 1400 because AVA and Josh committed acts of patent infringement, including the joint development and sale of the Infringing Remotes, in this District and AVA has its principal place of business in this District.

17. Finally, venue in this Court is also proper pursuant to a valid forum selection clause in the SDK Agreement between Snap One and Josh.ai.

GENERAL ALLEGATIONS

Snap One's Business

18. Snap One is in the business of designing, manufacturing, and distributing innovative residential and commercial electronics products, services, and software to authorized Dealers.

19. Snap One's product line includes its Control4 automation products, including control systems, lighting, interfaces (including remotes), audio and audiovisual distribution equipment, smart panels, applications, web services, and its proprietary operating system, which it sells exclusively through authorized Dealers.

20. Snap One is the successor-in-interest to Control4 Corporation. In 2019, Utah-based Control4 Corporation merged with North Carolina-based Wirepath Home Systems, LLC, dba SnapAV ("Wirepath"). Shortly thereafter and pursuant to an Assignment and Assumption Agreement, all Control4 Corporation's assets and liabilities, including its dealer agreements, were assigned to Wirepath.

21. In 2021, Wirepath changed its name to Snap One, LLC. Snap One continues to

maintain registered trademarks in the name “Control4” and to sell Control4-branded products to authorized Dealers.¹

22. Control4 branded products are connected devices that end user consumers use in homes and businesses as part of a smart control, automation, and media distribution system. A Control4 controller is central to each Control4 system and allows users to access, control, and manage their various connected devices through the Control4 web or mobile application, as well as from other in-home interfaces such as touch screens and remotes. A typical Control4 system is connected to multiple other devices, such as door locks, lights, speakers, intercoms, garage door openers, touch screens, audiovisual distribution equipment, network systems, power systems, and HVAC systems. As a result, with Control4 technology, the end-user can control and use all those proprietary and third-party products through one application. Control4 systems interoperate with thousands of third-party products.

23. Due, in part, to the complexity of designing and programming the Control4 and third-party products, and to ensure that the Control4 devices are integrated properly with other “smart” products, end users are required to have a Control4 authorized Dealer install and service a Control4 system. The Dealer installs and programs the system and, thereafter, provides ongoing support and maintenance. Programming may be completed on-site or, when authorized by the end user, remotely using Control4’s proprietary programming software.

24. Snap One maintains stringent requirements for its authorized Dealer program to protect its end users, brand, and reputation. Dealers that want to sell Control4 products and use

¹ In this Complaint, the entities of Control4 Corporation, WirePath, and Snap One are referred to collectively as “Snap One.”

Control4 software are required to complete a rigorous vetting process, a certification program, receive ongoing training, and operate under the terms of Authorized Dealer Agreements.

25. Snap One also licenses its Control4 software development tools, such as APIs, web services, and software code, to strategic partners who sign license agreements governing the use of such tools. These tools allow Snap One's partners to develop software applications and drivers that interact with Control4 products and systems.

26. Snap One limits access to its Control4 software and development tools to partners that sign license agreements and authorized Dealers that sign Authorized Dealer Agreements. Snap One also prohibits its partners and authorized Dealers from providing access to the Control4 software and systems to any unauthorized individuals or entities.

27. Violations of these agreements can cause significant monetary and reputational damages to Snap One.

Snap One and Josh.ai Enter into the SDK Agreement

28. On or about April 3, 2017, Snap One entered into an SDK Agreement with JStar, LLC, a former operating entity of Josh.ai.

29. Josh.ai is a company that develops and markets voice control technology that can be integrated into smart home devices and systems.

30. Pursuant to the SDK Agreement, Snap One granted Josh.ai a revocable license to use the Development Resources to develop, sell, and distribute application programs for Snap One's Control4 products and systems.

31. Importantly, the SDK Agreement prohibited Josh.ai from (among other things):

- a. Engaging (or enabling a third party to engage) in any activity that “interferes with, disrupts, damages, or accesses in an unauthorized manner any Control4 Products.”²; and
- b. Using (or enabling a third party to use) the Development Resources to “violate the legal rights (such as Intellectual Property Rights or rights of privacy and publicity) of others,” including Snap One.

32. The SDK Agreement also obligated Josh.ai to protect Snap One’s intellectual property—including trade secrets, software, code, and designs—contained within any of Josh.ai’s applications.

33. Pursuant to the SDK Agreement, Josh.ai released its first integration driver for Control4 in 2017, and Josh.ai and Snap One worked together thereafter to release subsequent versions of the integration for use by their joint customers.

Snap One Purchases NEEO and Develops an Upgraded Neeo Remote

34. On or about February 1, 2019, Snap One acquired all of the shares of NEEO, along with its intellectual property, trade secrets, and other confidential information.

35. At that time, NEEO was well-known in the CEDIA Industry for its development and distribution of the Neeo Remote, an innovative smart home remote and hub system.

36. NEEO began advertising and selling the Neeo Remote in 2015. In its marketing, NEEO prominently featured the remote’s “[a]ward winning simplicity and design.” The Neeo Remote received several notable design awards, including the 2015 Red Dot Award, the 2015 iF

² “Control4 Products” are defined in the SDK Agreement as “the collective reference to Control4’s hardware products, firmware, and software sold or provided by or through Control4.”

Design Award, and the 2017 Good Design Award. Its unique design, along with other aspects of the remote, propelled the Neeo Remote to become a one of the most sought-after remote solutions in the CEDIA Industry.

37. The Neeo Remote's popularity in the CEDIA Industry and its acclaimed, distinct, industrial design were primary reasons that Snap One purchased NEEO for millions of dollars. The NEEO intellectual property purchased by Snap One included patents and all other know-how, trade secrets, and other confidential information related to the Neeo Remote.

38. In connection with Snap One's acquisition of NEEO, Raphael Oberholzer, NEEO's former CEO and founder, and other NEEO leaders came to work for Snap One. Mr. Oberholzer signed an Employment Agreement with Snap One, in which he agreed to keep confidential and not use Snap One's trade secrets and other confidential information.

39. While at Snap One, Mr. Oberholzer led a team to develop and release a new and improved Neeo Remote that, although nearly identical in external design to the previous award-winning version, featured several internal technical and software upgrades to improve production and functionality. This upgraded Neeo Remote drew from and combined the intellectual property Snap One acquired from NEEO, intellectual property developed at Snap One after the NEEO acquisition, and Snap One's preexisting intellectual property (collectively, the "Intellectual Property").

40. Snap One released and began selling the upgraded Neeo Remote in 2019. And, since then, Snap One has continued to modify and enhance the Neeo Remote.

Mr. Oberholzer Leaves Snap One

41. Snap One and Mr. Oberholzer eventually agreed to terminate his Employment

Agreement, effective September 30, 2020. In his employment Termination Agreement (the “Termination Agreement”), Mr. Oberholzer reaffirmed his obligation to keep confidential all nonpublic information pertaining to Snap One and its business.

42. In conjunction with the Termination Agreement, Mr. Oberholzer and Snap One also executed a License Agreement (the “License Agreement”), which gave Mr. Oberholzer a non-exclusive, non-sublicensable license to the “Brain Assets.” The Brain Assets, as defined in Annex 1 of the License Agreement, consist of (among other things) the software, filesystems, and source code related to the control system associated with the Neeo Remote. They do not include the firmware, drivers, design, and other technologies related to the Neeo Remote itself.

43. The License Agreement provided that the Brain Assets could only be “held and used by one entity or person at a time” and prohibited Mr. Oberholzer from using the Brain Assets to “sell or market any products or services in direct competition with” Snap One for at least two years from the date of the agreement.

44. Snap One also entered into a Consultancy Agreement (the “Consultancy Agreement”) with Sichtfeld GmgH (“Sichtfeld”), a company owned by Mr. Oberholzer. Pursuant to the Consultancy Agreement, Sichtfeld agreed to provide specified services to Snap One, including services related to the design and development of the Neeo Remote.

45. The Consultancy Agreement provided, among other things, that any intellectual property, ideas, know-how, or technologies developed by Sichtfeld (with or without others) for Snap One belonged “entirely and exclusively” to Snap One.

46. The Consultancy Agreement further obligated Sichtfeld—and, by extension, Mr. Oberholzer—to keep confidential and not use any information obtained from or developed for Snap One during the consultancy.

47. The Consultancy Agreement terminated on March 31, 2021.

Mr. Oberholzer Launches AVA

48. Around that same time, Mr. Oberholzer co-founded a new company, AVA, with two other former Snap One employees. Together, these individuals recruited and hired other Snap One employees (collectively, the “Former Employees”) who worked with them on the development of the upgraded Neeo Remote.

49. These Former Employees signed certain agreements, including employment agreements and/or proprietary information and inventions agreements with Snap One, in which they, like Mr. Oberholzer, agreed to keep confidential and not use Snap One’s confidential and trade secret information.

50. At the end of 2021, Mr. Oberholzer contacted Snap One about his work with AVA. At that time, Mr. Oberholzer represented that AVA was involved in developing connected speakers, not a control system or smart home remote. In response, Snap One reiterated that Mr. Oberholzer was not allowed to be involved in a control system or remote company.

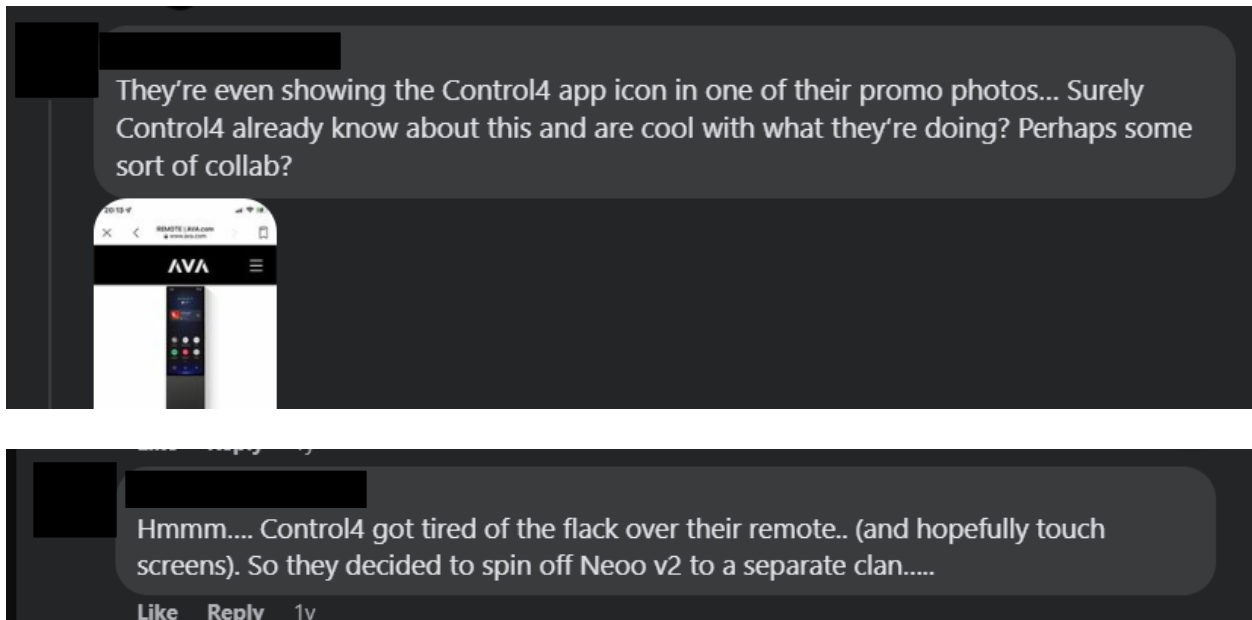
AVA Misappropriates the Neeo Remote Technology

51. Approximately two months later, AVA announced the release of the AVA Remote, a smart home remote that was strikingly similar to and directly competed with the Neeo Remote. Indeed, on information and belief, it appears that Mr. Oberholzer and the Former Employees took the Intellectual Property they worked with others to develop while employed at

Snap One (or NEEO) and used it to develop the AVA Remote. Likewise, on information and belief, it appears that Mr. Oberholzer improperly gave or sublicensed the Brain Assets to AVA to use in connection with the AVA Remote.

52. In addition to developing a rebranded version of the Neeo Remote, AVA also sought to trade on Snap One's reputation and goodwill by, among other things: (i) improperly using Snap One's Control4 trademarks on its website, in marketing hashtags, and in numerous images of the AVA Remote; (ii) representing that AVA was founded by "Former [Snap One] Executives"; and (iii) advertising the AVA Remote in closed Snap One Dealer groups. Snap One did not authorize or support the interoperation alleged between the AVA Remote and the Control4 platform.

53. AVA's actions created confusion in the CEDIA Industry about Snap One's ownership, support, or sponsorship of the AVA Remote:



54. Dealers also commented on the striking similarities between the remotes, saying

that the AVA remote “[l]ooks just like a neo [*sic*] remote,” and was a “just a Neeo 2.0.”

55. Given the market confusion and similarities between the remotes, Snap One sent AVA a letter in March 2022 demanding that it stop using Snap One’s Control4 trademarks and representing that the AVA Remote was compatible with Control4 products and systems. Snap One also expressed its belief that the AVA Remote improperly used and infringed on Snap One’s Intellectual Property and violated the contractual obligations of Mr. Oberholzer and the Former Employees.

56. After sending the letter, Snap One took steps to restrict AVA’s ability to integrate Control4 systems with the AVA Remote and AVA’s other products. After several communications between the parties, AVA stopped using the Control4 trademarks and announced that the AVA Remote was not authorized by or affiliated with Snap One.

57. Snap One reserved the right to take further action to protect its Intellectual Property and contractual rights as necessary.

Snap One and Josh.ai Enter into the Distributor Agreement

58. In August 2021, Snap One and Josh.ai decided to expand their relationship and entered into the Distributor Agreement.

59. The Distributor Agreement granted Snap One the exclusive right to sell certain Josh.ai products to Control4 Dealers in North America. It also obligated Josh.ai to refer all Control4 Dealers in North America to purchase its products through Snap One.

60. Relying on the exclusivity granted to it by the agreement and other commitments by Josh.ai, in or around the end of 2021, Snap One purchased millions of dollars’ worth of Josh.ai products to sell to Contol4 Dealers.

Josh Breaches and Undermines the Distributor Agreement

61. By the end of December 2022, Snap One's sales of Josh.ai products were significantly lower than expected. The disappointing results were the result of two main factors.

62. *First*, Josh.ai failed to refer all Control4 Dealers that desired to purchase Josh.ai products to Snap One. Instead, unbeknownst to Snap One, Josh.ai continued to market and sell its products to new and existing Control4 Dealers at prices that were lower than Snap One (as a reseller of the products) could offer.

63. Josh.ai's failure to refer new and existing Control4 Dealers to Snap One for Josh.ai products violated the "exclusive right" granted to Snap One in the Distributor Agreement. It also contravened Josh.ai's repeated assurances that Josh.ai was requiring Control4 Dealers, except for certain legacy Josh.ai Dealers that existed at the time of the contract, to purchase Josh.ai products through Snap One.

64. As a result of Josh.ai's breaches of the Distributor Agreement, from December 2021 to November 2022, Snap One lost hundreds of project sales, resulting in over \$1 million in lost revenue. At the same time, Josh.ai directly and materially benefited from Snap One's concentrated marketing efforts related to the partnership. Indeed, based on Snap One's available data, during the first year of the distribution partnership, more Control4 Dealers purchased Josh.ai products from Josh.ai than from Snap One.

65. *Second*, and relatedly, Josh.ai was marketing and selling the products directly to Control4 Dealers at prices well below the suggested reseller prices Snap One could offer. Josh.ai used these lower prices to circumvent its exclusivity obligations to Snap One and persuade Control4 Dealers to purchase Josh.ai products directly from Josh.ai.

66. Josh.ai also relaxed the requirements for Dealers to make direct purchases from Josh.ai and employed promotional pricing that widened the gap between the prices Snap One could offer and those being offered by Josh.ai. At times, this promotional pricing for Dealers was equal to or below the price Snap One paid for the Josh.ai products in its inventory, effectively forcing Snap One to indefinitely store the products or sell them at a loss.

67. Based on these discoveries, Snap One sent Josh.ai a letter on December 30, 2022. In that letter, Snap One demanded that Josh.ai abide by the terms of the agreement and buy back the Josh.ai products in Snap One's inventory that Snap One would have sold had Josh.ai properly required Control4 Dealers to purchase the products from Snap One.

68. In response, Josh.ai denied that it was marketing and selling to Control4 Dealers in violation of the Distribution Agreement and refused to accept the return of inventory. But less than three months later, and contradicting its assurances otherwise, Josh.ai announced a discounted purchasing relationship with ProSource, an industry leading buying group consisting primarily of Control4 Dealers. (See <https://www.residentialsystems.com/news/josh-ai-and-prosource-announce-partnership>.) This further violated the exclusivity provisions in the Distribution Agreement.

69. Having effectively prevented Snap One from selling its inventory, Josh.ai eventually offered to re-purchase the Snap One remaining Josh.ai inventory for approximately 50% less than what Snap One paid Josh.ai for the products. Snap One agreed to mitigate its damages.

Josh.ai and AVA Form a Partnership that Violates the SDK Agreement

70. While Josh was failing to refer Control4 Dealers to Snap One to purchase Josh.ai

products and undercutting Snap One on price, it was also exploring a potential partnership with AVA.

71. In the fall of 2022, Josh.ai and AVA partnered to jointly enable Josh.ai's mobile application and voice control system to run on the AVA Remote.

72. Pursuant to the SDK Agreement, Josh.ai had already integrated Snap One's Control4 platform into Josh.ai's system. Thus, by partnering with Josh.ai and integrating the AVA Remote with Josh.ai's system, AVA surreptitiously gained access to Snap One's Control4 platform—the very platform from which AVA was removed months earlier.

73. Then, in June 2023, Josh.ai announced the release of its own smart home remote—the Josh Remote. On information and belief, AVA provided the hardware, firmware, and industrial design for the Josh Remote. Both Josh.ai and AVA offer the Infringing Remotes for sale to Dealers.

74. Josh.ai and AVA have promoted the Infringing Remotes' ability to access Snap One's Control4 platform through Josh.ai's interface.

75. Josh.ai's distribution of the Josh Remote, and its role in enabling the AVA Remote to access Snap One's Control4 platform, violate the SDK Agreement and has caused (and continues to cause) Snap One significant harm.

Josh.ai and AVA Infringe on Snap One's Intellectual Property

76. In addition to violating the SDK Agreement, Josh.ai and AVA's partnership, joint development, and distribution and sale of the Infringing Remotes infringes on Snap One's valuable Intellectual Property, including Snap One's trade secrets, know-how, trade dress, and patents.

77. As alleged above, after working on teams that developed the upgraded version of the Neo Remote, Mr. Oberholzer and the Former Employees left Snap One and began working for AVA. On information and belief, in blatant disregard for their contractual obligations, Mr. Oberholzer and the Former Employees took the Intellectual Property they acquired or developed at Snap One (or NEEO) and used it to develop the AVA Remote.

78. Specifically, AVA incorporated or used many aspects of Snap One's Intellectual Property in the AVA Remote, including:

- a. Mechanical and circuitry concepts, software code and architecture, and materials incorporated into the Neo Remote;
- b. Development, sourcing, manufacturing methods and processes, quality control, and other business processes applied to the design and manufacture of the Neo Remote;
- c. Supplier, vendor, distributor, and other strategic partner information involved in the design, production, and distribution of the Neo Remote;
- d. Products, concepts, systems, or services covered by Snap One's Patents, as further discussed herein; and
- e. Unique and distinctive design characteristics of the Neo Remote.

79. AVA also improperly used the Brain Assets in connection with the Remote, which it knew or should have known (through Mr. Oberholzer) were non-sublicensable and could not be used to sell or market a product in direct competition with Snap One.

80. After Snap One removed AVA's ability to integrate the Control4 platform into its products, AVA looked for unauthorized ways to access the platform. To that end, it appears that

AVA approached Josh.ai, which had a license to integrate Control4 systems into its applications and interface, about a potential collaboration.

81. On information and belief, Josh.ai saw the collaboration as an opportunity to bypass years of work and intellectual capital needed to develop an innovative smart remote like the Neeo Remote. Accordingly, on information and belief, Josh.ai used Snap One's Intellectual Property improperly incorporated in the AVA Remote and, in conjunction with AVA, developed the Josh Remote to sell to Dealers.

82. Josh.ai knew or should have known that the AVA Remote improperly incorporated Snap One's Intellectual Property when it began its collaboration with AVA. For example, as shown below, the Neeo Remote features a unique and distinctive design that is well-known in the CEDIA Industry. Josh.ai knew or should have known that AVA misappropriated Snap One's Intellectual Property by looking at the AVA Remote's copycat design.



83. Indeed, after AVA released the AVA Remote, many in the CEDIA Industry suspected that AVA had improperly copied the Neo Remote. Further, the industry knew that Snap One purchased NEEO and that Mr. Oberholzer and the Former Employees worked for Snap One before starting AVA. Those in the industry also knew that AVA's access to the Control4 platform had been restricted in the spring of 2022 because the AVA Remote no longer worked with Snap One's Control4 smart home platform.

84. Finally, on information and belief, Josh.ai (and AVA) had actual or constructive knowledge of Snap One's patents, given Snap One's position as a leader in the CEDIA Industry and its publicized acquisition of NEEO.

85. Despite this knowledge (or willful ignorance), Josh.ai moved forward on its partnership with AVA, jointly developed the Josh Remote, and profited and continues to profit

from that partnership and its distribution of the Josh Remote.

Defendants' Infringement of Snap One's Patents

86. Snap One is a leader in designing, manufacturing, and distributing innovative home and commercial electronics products, services, and software to authorized Dealers. Certain product development and research divisions of Snap One date back to as early as 1981.

87. When Control4 Corporation first launched in 2003, it revolutionized the automation and control business, providing whole home control. Early product innovations at that time included remote and touch screen interface devices, control systems, audio and media distribution, and certain networked solutions, as shown below.



First Generation Control4 Products

88. Since then, Control4 Corporation (now Snap One) has continued to be at the forefront of innovation and offer cutting-edge technology for its customers' homes and businesses.

89. Snap One's products and services allow end-user customers to, among other things, control and manage network, security, communications, multi-room audio, AV distribution, power, smart lighting, electromechanical, HVAC, surveillance, and numerous other types of electronics systems with products such as control systems, applications, touch screens, keypads, and remotes. Snap One also supports integration with tens of thousands of third-party products to achieve a high level of whole home control.

90. Snap One's patented innovations cover a breadth of areas including control system processes and interfaces, network topology and architecture, mesh networking, remote management software, electrical circuitry design, lighting control, efficient power distribution, energy management, audio playback and distribution, privacy and data protection, keypad and light switch mechanical design, and interface device design.

91. In addition to Snap One's actual development and sale of automation and control products, Snap One has obtained over one hundred patents on these systems. These patents are the result of years of innovation, investment, and hard work that helped Snap One become a leading provider of custom smart-home, control, and automation solutions.

92. Many of Snap One's technological innovations and acquisitions related to its control, automation, remote, audio distribution, and network technologies are protected by, *inter alia*, a portfolio of utility and design patents, including United States Patent Nos. 10,877,623 ("623 Patent"), 8,375,137 ("137 Patent"), 11,140,426 ("426 Patent"), 7,866,338 ("338 Patent"), and 10,756,984 ("984 Patent") (collectively, the "Snap One Patents"). Copies of the Snap One Patents are attached as Exhibits A through E.

93. Snap One is the owner of the Snap One Patents and holds all rights, title, and interest in and to the Snap One Patents, including the right to sue and recover for all past, present, and future infringements.

94. Snap One has not authorized, licensed, or otherwise given permission for Defendants to use any of the Snap One Patents or to make, use, offer for sale, or sell any product or service under any claim of the Snap One Patents.

95. Snap One's presence in the market and its portfolio of patents was well-known to Mr. Oberholzer and the Former Employees, all of whom joined AVA after working for Snap One.

96. As NEEO's former CEO, Mr. Oberholzer was also well-aware of the intellectual property and products Snap One acquired from NEEO and Snap One's other intellectual property, especially its remote technology. Indeed, after Snap One's acquisition of NEEO, Mr. Oberholzer (along with the Former Employees) worked on an upgraded version of the Neo Remote.

97. The Neo Remote incorporates and relies on technologies protected by at least some of the Snap One Patents. Thus, on information and belief, in working to upgrade the Neo Remote, Mr. Oberholzer and the Former Employees were aware of at least some of the Snap One Patents.

98. AVA makes, uses, sells, offers for sale, the AVA Remote, AVA Streamer, and AVA Speakers, and other integrated devices, which are shown on <https://www.ava.com/> (collectively, the "AVA Products").

99. Josh makes, uses, sells, or offers for sale the Josh Remote, Josh Core, Josh Micro, Josh Nano, and other integrated devices, which are shown on <https://www.josh.ai/> (collectively, the “Josh Products”).

100. As detailed more specifically below, AVA and Josh have and continue to directly and indirectly infringe one or more claims of each of the Snap One Patents by making, using, selling, offering for sale in the United States, and/or importing into the United States, the AVA Products and Josh Products to their customers.

Snap One Contacts Josh.ai Regarding Its Misconduct

101. On July 14, 2023, Snap One sent Josh.ai a letter. The letter informed Josh.ai that its partnership with AVA and distribution of the Josh Remote violated the SDK Agreement and infringed on Snap One’s Intellectual Property.

102. The letter demanded that Josh.ai: (1) come into compliance with the SDK Agreement, (2) stop providing the AVA Remote and other AVA products with access to the Control4 platform and Snap One’s supporting services, and (3) discontinue the use, marketing, distribution, and sale of the Josh Remote. Snap One gave Josh.ai fourteen (14) days to comply with these demands.

103. Over the few weeks, Snap One and Josh.ai exchanged communications, but Josh.ai did not agree to stop providing AVA with unauthorized access to the Control4 platform, nor did it agree to stop distributing and selling the Josh Remote.

104. Instead, Josh.ai continues to partner with AVA, and both companies continue to jointly develop, market, distribute, and sell the Infringing Remotes in violation of their contractual and legal duties to Snap One, forcing Snap One to commence this lawsuit.

FIRST CLAIM FOR RELIEF

(Breach of Contract (SDK Agreement) – Against Josh.ai)

105. Snap One incorporates by reference the allegations set forth in the preceding paragraphs as though set forth fully herein.

106. The SDK Agreement is a valid and enforceable contract between Snap One and Josh.ai.

107. Snap One performed its obligations under the SDK Agreement by acting in accordance with the terms set forth therein.

108. The SDK Agreement prohibited Josh.ai from (among other things):

- a. Engaging (or enabling a third party to engage) in any activity that “interferes with, disrupts, damages, or accesses in an unauthorized manner any Control4 Products”; and
- b. Using (or enabling a third party to use) the Development Resources to “violate the legal rights (such as Intellectual Property Rights or rights of privacy and publicity) of others,” including Snap One.

109. The SDK Agreement also obligated Josh.ai to protect Snap One’s intellectual property—including trade secrets, software, code, and designs—contained within any of Josh.ai’s applications.

110. Josh.ai materially breached the SDK Agreement by, among other things:

- a. Facilitating AVA’s access to the Control4 platform in an unauthorized manner;
- b. Marketing, distributing, and selling the Josh Remote, which infringes on Snap One’s Intellectual Property, to Dealers; and

- c. Failing to protect Snap One’s Control4 platform—which is integrated into Josh.ai’s applications—from unauthorized access.

111. As a direct and proximate result of Josh.ai’s breaches of the SDK Agreement, Snap One has suffered, and continues to suffer, damages in an amount to be determined at trial but not less than the jurisdictional amount.

SECOND CLAIM FOR RELIEF

(Misappropriation of Trade Secrets in Violation
of the DTSA (18 U.S.C. § 1836) – Against All Defendants)

112. Snap One incorporates by reference the allegations set forth in the preceding paragraphs as though fully set forth herein.

113. Snap One’s Intellectual Property includes certain trade secrets within the definition of 18 U.S.C. § 1839(3). These trade secrets include, among other things: (1) mechanical and circuitry concepts, software code and architecture, and materials incorporated into the Neeo Remote; (2) development, sourcing, manufacturing methods and processes, quality control, and other business processes applied to the design and manufacture of the Neeo Remote; and (3) supplier, vendor, distributor, and other strategic partner information involved in the design, production, and distribution of the Neeo Remote.

114. Snap One has taken reasonable measures to protect these concepts, processes, methods, and techniques by securing them in limited access business and engineering repositories that are secured by cyber, informational, and/or physical protections designed by security experts. Snap One also protects these trade secrets by requiring employees, personnel, and development and manufacturing service providers with access to the trade secrets to sign

contracts agreeing that they will keep confidential and take steps to protect the information from being disclosed.

115. The concepts, processes, methods, and techniques also derive independent economic value, both actual and potential, from not being generally known to, and not being readily ascertainable through proper means by, another person who can obtain economic value from the disclosure or use of the information. Among other things, the trade secret information allows for and promotes unique design aesthetics, reduced supply and design timelines and costs, effective distribution, and increased product performance in the areas of useability, quality, energy efficiency, and added features.

116. AVA acquired Snap One's trade secrets through Mr. Oberholzer and the Former Employees. As alleged above, after Snap One purchased NEEO and all intellectual property related to the Neo Remote, Mr. Oberholzer came to work for Snap One. At Snap One, Mr. Oberholzer and the Former Employees worked on a team to develop an upgraded version of the Neo Remote, which incorporated the intellectual property purchased from NEEO and Snap One's preexisting intellectual property and confidential information.

117. On information and belief, when Mr. Oberholzer and the Former Employees left Snap One and started AVA, they took Snap One's trade secrets with them. And, in violation of their various agreements with Snap One, Mr. Oberholzer and the Former Employees used Snap One's trade secrets to develop the AVA Remote for AVA.

118. Josh.ai acquired or used Snap One's trade secrets when it partnered with AVA in the fall of 2022, embedded Josh.ai's applications and remote interface within the AVA Remote,

and (with AVA) developed the Josh Remote. At that time, Josh.ai knew, or had reason to know, that the Infringing Remotes improperly incorporate Snap One's trade secrets.

119. At the very least, Josh.ai knew that the Infringing Remotes improperly incorporate Snap One's trade secrets when Snap One informed it of that fact on July 14, 2023.

120. Defendants continue to use Snap One's trade secrets to manufacture, distribute, and sell the Infringing Remotes.

121. As a direct and proximate result of Defendants' misappropriation of its trade secrets, Snap One has suffered, and continues to suffer, damages in an amount to be determined at trial.

122. Further, Defendants' misappropriation and use of Snap One's trade secrets is willful and malicious. Through Mr. Oberholzer and the Former Employees, AVA was aware of the value of Snap One's trade secrets and the harm it would cause Snap One if they were misappropriated. Nevertheless, AVA used the trade secrets to develop its business and produce a nearly identical remote to the Neeo Remote. Josh.ai became aware of AVA's misconduct, at the latest, in July 2023. Thus, Defendants' conduct is willful and malicious, and the Court may award Snap One multiple damages and attorney fees under 18 U.S.C. §§ 1836(b)(3)(C) and 1836(b)(3)(D).

THIRD CLAIM FOR RELIEF

(Misappropriation of Trade Secrets in Violation of the Utah's Uniform Trade Secrets Act (Utah Code §§ 13-24-1 *et seq.*) – Against All Defendants)

123. Snap One incorporates by reference the allegations set forth in the preceding paragraphs as though fully set forth herein.

124. Snap One's Intellectual Property includes certain trade secrets within the definition of Utah Code § 13-24-2(4). These trade secrets include, among other things: (1) mechanical and circuitry concepts, software code and architecture, and materials incorporated into the Neeo Remote; (2) development, sourcing, manufacturing, quality control, and other business processes applied to the design and manufacture of the Neeo Remote; and (3) supplier, vendor, distributor, and other strategic partner information involved in the design, production, and distribution of the Neeo Remote.

125. Snap One has taken reasonable measures to protect these concepts, processes, methods, and techniques by securing them in limited access business and engineering repositories that are secured by cyber, informational, and/or physical protections designed by security experts. Snap One also protects these trade secrets by requiring employees, personnel, and development and manufacturing service providers with access to the trade secrets to sign contracts agreeing that they will keep confidential and take steps to protect the information from being disclosed.

126. The concepts, processes, methods, and techniques also derive independent economic value, both actual and potential, from not being generally known to, and not being readily ascertainable through proper means by, another person who can obtain economic value from the disclosure or use of the information. Among other things, the trade secret information allows for and promotes unique design aesthetics, reduced supply and design timelines and costs, effective distribution, and increased product performance in the areas of useability, quality, energy efficiency, and added features.

127. AVA acquired Snap One's trade secrets through Mr. Oberholzer and the Former Employees. As alleged above, after Snap One purchased NEEO and all intellectual property related to the Neeo Remote, Mr. Oberholzer came to work for Snap One. At Snap One, Mr. Oberholzer and at least some of the Former Employees worked on a team to develop an upgraded version of the Neeo Remote, which incorporated the intellectual property purchased from NEEO and Snap One's preexisting intellectual property and confidential information.

128. On information and belief, when Mr. Oberholzer and the Former Employees left Snap One and started AVA, they took Snap One's trade secrets with them. And, in violation of their various agreements with Snap One, Mr. Oberholzer and the Former Employees used Snap One's trade secrets to develop the AVA Remote for AVA.

129. Josh.ai acquired or used Snap One's trade secrets when it partnered with AVA in the fall of 2022, embedded Josh.ai's applications and remote interface within the AVA Remote, and (with AVA) developed the Josh Remote. At that time, Josh.ai knew, or had reason to know, that the Infringing Remotes improperly incorporate Snap One's trade secrets.

130. At the very least, Josh.ai knew that the Infringing Remotes improperly incorporate Snap One's trade secrets when Snap One informed it of that fact on July 14, 2023.

131. Defendants continue to use Snap One's trade secrets to manufacture, distribute, and sell the Infringing Remotes.

132. As a direct and proximate result of Defendants' misappropriation of its trade secrets, Snap One has suffered, and continues to suffer, damages in an amount to be determined at trial.

133. Further, Defendants' misappropriation and use of Snap One's trade secrets is willful and malicious. Through Mr. Oberholzer and the Former Employees, AVA was aware of the value of Snap One's trade secrets and the harm it would cause Snap One if they were misappropriated. Nevertheless, AVA used the trade secrets to develop its business and develop a nearly identical remote to the Neeo Remote. Josh.ai became aware of AVA's misconduct, at the latest, in July 2023. Thus, Defendants' conduct is willful and malicious, and the Court may award Snap One multiple damages and attorney fees under 18 U.S.C. §§ 1836(b)(3)(C) and 1836(b)(3)(D).

FOURTH CLAIM FOR RELIEF

(Trade Dress Infringement in Violation of Section 43(a) of the Lanham Act –
Against All Defendants)

134. Snap One incorporates by reference the allegations set forth in the preceding paragraphs as though fully set forth herein.

135. The Neeo Remote has a distinct overall look and feel stemming from, at least, its: (1) machined aluminum housing in anodized black or silver, (2) machined LCD glass fit in an ultra-thin aluminum housing bezel, (3) a touchscreen that includes a maximally active screen area that adjoins a barely visible bezel, (4) a circular docking station composed of inlaid machine aluminum with a magnetic array and added weight, and (5) an exterior shape with its front, back, top, and sides straight and flat with minimal corner radius, and a bottom side with a rounded edge that fits directly into rounded dock opening. The distinct look has remained consistent since the Neeo Remote was first announced for presale in January 2015 until the present day, and Dealers in the CEDIA Industry identify this distinct look and feel with Snap One.

136. The combination of these nonfunctional elements, among others, and the total impression of the Neo Remote is sufficiently distinct to Dealers in the CEDIA Industry, such that this trade dress identifies Snap One as the source of the Neo Remote. Within the applicable market for control system remotes, the CEDIA Industry, there are roughly five remote control solutions. As shown below, the Neo Remote is sufficiently distinct such that none of the aforementioned elements of look and feel have been replicated in any of the other competing remote solutions.

CEDIA Control System Market Competing Remote Solutions



Crestron
TSR-310



ELAN
HR10



RTI
T2i



Savant
X2

137. Snap One has extensively marketed and advertised the Neeo Remote to Dealers within the CEDIA Industry. Snap One conservatively estimates that it has spent over \$5 million annually in direct marketing to promote the Neeo Remote and other Control4-branded products.

138. Snap One's efforts helped the Neeo Remote become one of the primary remote solutions in the CEDIA Industry.

139. Through its extensive and continuous use and promotion, the Neeo Remote's trade dress has become a well-known indicator of the origin and modern state-of-the-art quality of the Neeo Remote and has acquired secondary meaning. This distinct trade dress acquired secondary meaning before AVA and Josh.ai commenced their unlawful infringement.

140. Through their production and sale of the Infringing Remotes, Defendants have intentionally, knowingly, deliberately, and willfully infringed (and continue to infringe) Snap One's trade dress rights through their blatant imitation of the Neeo Remote in interstate commerce. This intentional, wrongful imitation includes, but is not limited to: (1) machined aluminum housing in anodized black or silver, (2) machined LCD glass fit in an ultra-thin aluminum housing bezel, (3) a touchscreen that includes a maximally active screen area that adjoins a barely visible bezel, (4) a circular docking station composed of inlaid machine aluminum with a magnetic array and added weight, and (5) an exterior shape with its front, back, top, and sides straight and flat with minimal corner radius, and a bottom side with a rounded edge that fits directly into rounded dock opening.

141. The result, as shown below, is that the look and feel of the Infringing Remotes are virtual copycats of the Neo Remote.



142. The Infringing Remotes have caused and are likely to continue causing confusion, mistake, or deception as to the source, affiliation, connection, or association of Infringing Remotes with Snap One. This includes, for example, the confusion, mistake, or deception that creates initial customer interest in the Infringing Remotes.

143. As a result of AVA's infringement of Snap One's trade dress, Snap One has suffered substantial harm, including without limitation harm to its reputation, loss of business goodwill, and lost sales. Snap One has suffered, and continues to suffer, damages in an amount to be determined at trial.

144. Snap One has also suffered and continues to suffer immediate and irreparable injury for which it has no adequate remedy at law. Snap One is therefore entitled to injunctive

relief and up to three times its actual damages and/or an award of Defendants' profits, as well as costs and Snap One's reasonable attorney fees under 15 U.S.C. §§ 1116–17.

FIFTH CLAIM FOR RELIEF

(Infringement of U.S. Patent No. 10,877,623 – Against All Defendants)

145. Snap One incorporates by reference the allegations set forth in the preceding paragraphs as though fully set forth herein.

146. The '623 Patent was filed on June 13, 2008, and was duly and legally issued by the U.S. Patent and Trademark Office on December 29, 2020, entitled, "Dynamic interface for remote control of a home automation network."

147. Snap One is the owner by assignment of the '623 Patent. A copy of the '623 Patent is attached hereto as Exhibit A. Snap One holds all rights, title, and interest to the '623 Patent, including the sole and exclusive right to bring a claim for its infringement.

148. The '623 Patent is valid, enforceable, and was duly issued in full compliance with Title 35 of the United States Code.

149. The '623 Patent is generally directed to a "remote control for a home automation network."

150. The '623 Patent teaches technology improvements regarding the usability and performance of a smart remote control for a home automation system that efficiently connects and controls a variety of devices, such as lights, speakers, climate control, and audio/visual equipment. This technology improves an end user's control of her smart-home devices, for example, by dynamically configuring and remapping the displayed elements and selection operations for a user interface of the remote control as the end user selects or scrolls between

device options on the remote control. The inventions described in the '623 Patent represent improvements in the art.

151. AVA and Josh.ai manufacture (or have manufactured), sell and offer to sell, among other things, smart-home remotes that are integrated with a home automation network and products. Such products include the AVA Remote and the Josh.ai Remote, as well as components thereof.

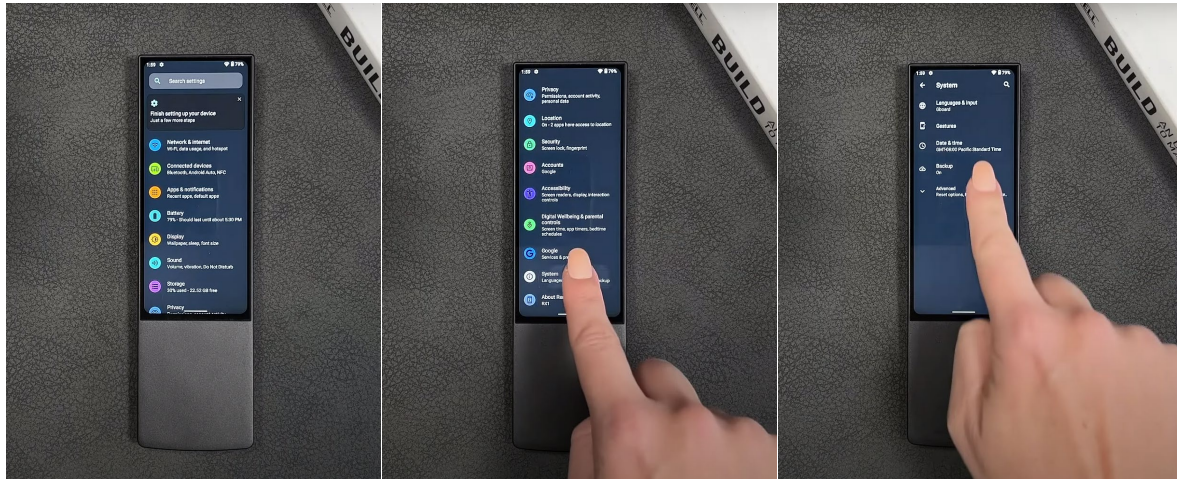
152. The "'623 Infringing Devices" includes at least any of the Infringing Remotes, as defined herein which comprises of the AVA Remote and the Josh Remote, as well as the Josh Core, as depicted at <https://www.josh.ai/core/>.

153. AVA and Josh.ai import, make, use, sell, and/or offer to sell the '623 Infringing Devices, as well as components including accessories and associated software for use with the '623 Infringing Devices to provide access and integration to home automation networks that connect to a variety of devices such as lights, speakers, climate control, and audio/visual equipment, where, on information and belief, the different automation devices have different control options that use respective mappings of the user input operations as a user scrolls and selects between different automation devices on the user interface of the '623 Infringing Devices.

154. Defendants are liable for their infringement of the '623 Patent pursuant to 35 U.S.C. §§ 271(a), (b), and (c).

155. The '623 Infringing Devices are covered, either literally or under the doctrine of equivalents, by one or more claims of the '623 Patent. For example, at a minimum, the '623 Infringing Devices connect to a home automation network in a manner that embodies one or more claimed inventions in the '623 Patent, including at least claim 24. Some examples of the

AVA Remote and Josh Remote for a home automation network for with control operations and mapping is shown below:



https://www.youtube.com/watch?v=bF98PcvA5ec&t=6s&ab_channel=SmashTech



<https://www.ava.com/remote>



[https://youtu.be/zHWSjLvnKRg?t=694.](https://youtu.be/zHWSjLvnKRg?t=694)

156. As shown in the examples above and further described below, the '623 Infringing Devices meet the limitations of claim 24.

157. For example, the '623 Infringing Devices and associated components/software include a display device that displays a menu to a user that can be used with an automation controller.

158. The '623 Infringing Devices and associated components/software include a transceiver for communicating with the remote. The '623 Infringing Devices and associated components/software further include memory capacity and present menu items from the automation controller. The user can navigate through the menu items by scrolling and selecting menu items.

159. The '623 Infringing Devices and associated components/software include the ability to communicate with the automation controller to control lighting dimmers and switches, audio/visual equipment, and climate control in response to control messages from the remote. *See e.g.*, <https://www.ava.com/featured-partners>.

160. Defendants conduct, as set forth herein, have directly infringed and continue to directly infringe one or more claims of the '623 Patent, including at least claim 24, by making, using, selling, offering for sale within the United States, and/or importing into the United States products, systems, and/or services, in violation of at least 35 U.S.C. §§ 271(a) and 281 *et seq.*

161. Additionally, Defendants have indirectly infringed and continue to indirectly infringe one or more of the claims of the '623 Patent, in violation of 35 U.S.C. § 271(b), by knowing and intentionally inducing infringement of the '623 Patent, literally and/or under the doctrine of equivalents.

162. Defendants have and continue to actively induce their inducing its customers to sell and/or use the '623 Infringing Devices in a manner directly infringing one or more claims of the '623 Patent, including at least claim 24. On information and belief, (a) Defendants have had actual or constructive knowledge of the '623 Patent, or at least have actual knowledge as of filing of this Complaint, (b) Defendants intentionally cause, urge, or encourage users of the '623 Infringing Devices to directly infringe one or more claims of the '623 Patent by, for example, promoting, advertising, and instructing customers and potential customers about the '623 Infringing Devices and uses thereof, including infringing uses, (c) Defendants know (or should know) that their actions will induce customers/users of the '623 Infringing Devices to directly infringe one or more claims the '623 Patent, and (d) customers/users of '623 Infringing Devices directly infringe one or more claims of the '623 Patent.

163. In addition, Defendants have supplied and continue to supply the '623 Infringing Devices to customers while knowing that use (and inducing such use) of the '623 Infringing Devices will infringe one or more claims of the '623 Patent, and that Defendants' customers then directly infringe one or more claims of the '623 Patent by selling and/or using the '623 Infringing Devices.

164. Additionally, Defendants have indirectly infringed and continue to indirectly infringe one or more of the claims of the '623 Patent, in violation of 35 U.S.C. § 271(c), by offering to sell or selling within the United States, or importing into the United States, components in connection with the '623 Infringing Devices that contribute to the direct infringement of the '623 Patent by customers/users of the '623 Infringing Devices.

165. On information and belief, (a) Defendants have actual or constructive knowledge of the '623 Patent, or at least have knowledge at the time of this suit, (b) Defendants offer or sale, sell, or import Infringing Remote products that comprise one or more material components of the invention of the '623 Patent and are not staple articles of commerce suitable for substantial non-infringing use, (c) Defendants know (or should know) that such component(s) are especially made or especially adapted for use in an infringement of the '623 Patent, and (d) customers/users of the '623 Infringing Devices comprise such material component(s) directly infringe one or more claims of the '623 Patent. For example, at a minimum, Defendants offer for sale, sell, and/or import '623 Infringing Devices that comprise software and/or hardware components adapted for use in one or more inventions claimed in the '623 Patent, and Defendants' customers then directly infringe one or more claims of the '623 Patent by selling and/or using the '623 Infringing Devices that comprise these components.

166. AVA and Josh.ai import, make, use, sell, and/or offer to sale the '623 Infringing Devices, despite an objectively high likelihood that their actions constitute infringement.

167. Snap One has satisfied the statutory requirements to collect pre-suit damages for infringement of each of the Snap One Patents, including, without limitation, any applicable provisions of 35 U.S.C. § 287, by, at least, providing notice via a website made available at <https://www.control4.com/legal/patents/>.

168. Defendants' acts of infringement have caused damage to Snap One, and Snap One is entitled to recover the damages sustained because of Defendants' wrongful acts including lost profits but in no event less than a reasonable royalty in an amount to be determined including any further relief, such as compensatory or equitable, as deemed proper and just by the Court.

169. Snap One is also entitled to injunctive relief because Defendants' infringement of the '623 Patent has and will continue to irreparably harm Snap One's business, for which there is no adequate remedy at law.

170. Defendants knew at least as early as the filing of this Complaint that, through its encouragement, the '623 Infringing Devices would result in infringement of one or more claims of the '623 Patent.

171. Despite having actual knowledge, Defendants have continued to carry out infringing activities. Defendants have had actual knowledge, or at least constructive knowledge, of the '623 Patent. Defendants have known, or have been willfully blind, to the existence of the '623 Patent and associated infringement thereof.

172. Defendants have had notice and have willfully infringed or continue to willfully infringe the '623 Patent as of the filing of this Complaint, entitling Snap One to increased damages under 35 U.S.C. § 284 and to attorney fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

173. By reason of the foregoing, Snap One is entitled to relief against Defendants, pursuant to at least 35 U.S.C. §§ 283–85.

SIXTH CLAIM FOR RELIEF

(Infringement of U.S. Patent No. 8,375,137 – Against AVA)

174. Snap One incorporates by reference the allegations set forth in the preceding paragraphs as though fully set forth herein.

175. The '137 Patent was filed on July 22, 2009, and was duly and legally issued by the U.S. Patent and Trademark Office on Feb. 12, 2013, entitled, "System and method for

streaming audio using a send queue.” Snap One holds all rights, title, and interest to the ’137 Patent, including the sole and exclusive right to bring a claim for its infringement.

176. Snap One is the owner by assignment of the ’137 Patent. A copy of the ’137 Patent is attached hereto as Exhibit B.

177. The ’137 Patent is valid, enforceable, and was duly issued in full compliance with Title 35 of the United States Code.

178. The ’137 Patent is generally directed to a streaming audio that is “played simultaneously across multiple rooms.”

179. The ’137 Patent teaches technology improvements addressing and overcoming problems with streaming audio to multi-speaker rooms or room systems, for example, such that a client device providing media playback can receive a pre-buffered amount of data of the media stream to fill the media client’s user buffer upon an activation event, thereby improving the art.

180. AVA manufactures and sells, among other things, streaming systems that integrate with speakers.

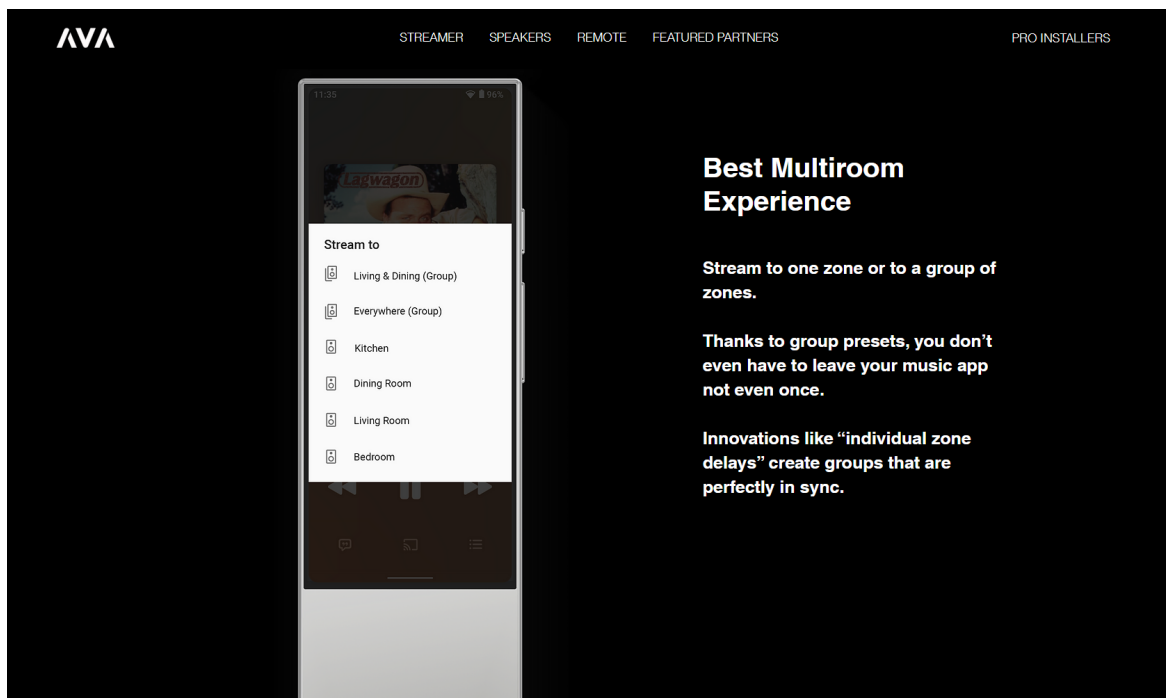
181. The “AVA System” comprises at least, any of the AVA Remote as defined herein, AVA Streamer as described at <https://www.ava.com/streamer>, and the AVA Speakers, as described at <https://www.ava.com/speakers>.

182. On information and belief, AVA imports, makes, uses, sells, and/or offers to sell AVA Devices, as well as components (which further includes any accessories) including software associated therewith and for use with the AVA System to provide access and integration to AVA Speakers streaming audio to multi-speaker rooms or room systems, such that

a speaker can receive a pre-buffered amount of data of the media stream to fill the media client's user buffer upon an activation event.

183. AVA is liable for its infringement of the '137 Patent pursuant to 35 U.S.C. §§ 271(a), (b), and (c).

184. The AVA System is covered, either literally or under the doctrine of equivalents, by one or more claims of the '137 Patent. For example, at a minimum, the AVA System connects to a system for streaming audio to multiple speakers in a manner that embodies one or more claimed inventions in the '137 Patent, including at least claim 1. Some examples of the system for streaming audio to multiple speakers is shown below:



<https://www.ava.com/streamer>

185. As shown above and described below, the AVA System meets the limitations of claim 1. For example, the AVA System and associated components/software includes a streaming media system configured to prepare to stream media, such as audio.

186. The AVA System and associated components/software include a media server, such as the AVA Streamer, to send out media packets for an audio media stream.

187. On information and belief, the AVA System and associated components/software include a framer in the media server to divide the frames of a media stream into packets. The packets can be received by the media client, such as the AVA Speaker, for example, to playback the frames of the media stream received from the media server.

188. On information and belief, the AVA System and associated components/software includes a speaker to store an amount of the packetized media stream provided by the AVA Streamer to fill the speaker's buffer by transferring multiple frames from a send queue of the AVA streamer in a single packet.

189. AVA's conduct, as set forth herein, has directly infringed and continues to directly infringe one or more claims of the '137 Patent, including at least claim 1, by making, using, selling, offering for sale within the United States, and/or importing into the United States products, systems, and/or services, in violation of at least 35 U.S.C. §§ 271(a) and 281 *et seq.*

190. Additionally AVA has indirectly infringed and continues to indirectly infringe one or more of the claims of the '137 Patent, in violation of 35 U.S.C. § 271(b), by knowing and intentionally inducing infringement of the '137 Patent, literally and/or under the doctrine of equivalents. AVA has and continues to actively induce their customers to sell and/or use the AVA System in a manner directly infringing one or more claims of the '137 Patent, including at least claim 1.

191. On information and belief, (a) AVA has had actual or constructive knowledge of the '137 Patent, or at least has actual knowledge as of the filing of this Complaint, (b) AVA

intentionally causes, urges, or encourages users of the AVA System to directly infringe one or more claims of the '137 Patent by, for example, promoting, advertising, and instructing customers and potential customers about the AVA System and uses thereof, including infringing uses, (c) AVA knows (or should know) that its actions will induce customers/users of the AVA System to directly infringe one or more claims of the '137 Patent, and (d) customers/users of AVA System directly infringe one or more claims of the '137 Patent. In addition, AVA has supplied and continues to supply the AVA System to customers while knowing that use (and inducing such use) of the AVA System will infringe one or more claims of the '137 Patent, and that AVA's customers then directly infringe one or more claims of the '137 Patent by selling and/or using the AVA System.

192. Additionally, AVA has indirectly infringed and continues to indirectly infringe one or more of the claims of the '137 Patent, in violation of 35 U.S.C. § 271(c), by offering to sell or selling within the United States, or importing into the United States, components in connection with the AVA System that contribute to the direct infringement of the '137 Patent by users of the AVA System.

193. On information and belief, (a) AVA has actual or constructive knowledge of the '137 Patent, or at least have knowledge at the time of this suit, (b) AVA offers or sales, sells, or imports the AVA System products that comprise one or more material components of the invention of the '137 Patent and are not staple articles of commerce suitable for substantial noninfringing use, (c) AVA knows (or should know) that such component(s) are especially made or especially adapted for use in an infringement of the '137 Patent, and (d) customers/users of the AVA System comprise such material component(s) directly infringe one or more claims of

the '137 Patent. For example, at a minimum, AVA offers for sale, sells, and/or imports the AVA System that comprises software and/or hardware components adapted for use in one or more inventions claimed in the '137 Patent, and AVA's customers then directly infringe one or more claims of the '137 Patent by selling and/or using the AVA System that comprise these components.

194. AVA imports, makes, uses, sells, and/or offers to sale the AVA System, despite an objectively high likelihood that their actions constitute infringement.

195. Snap One has satisfied the statutory requirements to collect pre-suit damages for infringement of each of the Snap One Patents, including, without limitation, any applicable provisions of 35 U.S.C. § 287, by, at least, providing notice via a website made available at <https://www.control4.com/legal/patents/>.

196. AVA's acts of infringement have caused damage to Snap One, and Snap One is entitled to recover the damages sustained because AVA's wrongful acts including lost profits but in no event less than a reasonable royalty in an amount to be determined including any further relief, such as compensatory or equitable, as deemed proper and just by the Court.

197. Snap One is also entitled to injunctive relief because AVA's infringement of the '137 Patent will continue to damage Snap One's business, causing irreparable harm, for which there is no adequate remedy at law.

198. AVA knew at least as early as the filing of this Complaint that, through its encouragement, the '137 AVA System would result in infringement of one or more claims of the '137 Patent.

199. Despite having actual knowledge, AVA has continued to carry out infringing activities. AVA has had actual knowledge, or at least constructive knowledge, of the '137 Patent. AVA has known, or have been willfully blind, to the existence of the '137 Patent and associated infringement thereof.

200. AVA has had notice and has willfully infringed and continues to willfully infringe as of the filing of this Complaint the '137 Patent, entitling Snap One to increased damages under 35 U.S.C. § 284 and to attorney fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

201. By reason of the foregoing, Snap One is entitled to relief against AVA pursuant to at least 35 U.S.C. §§ 283–85.

SEVENTH CLAIM FOR RELIEF

(Infringement of U.S. Patent No. 11,140,426 – Against Josh.ai)

202. Snap One incorporates by reference the allegations set forth in the preceding paragraphs as though fully set forth herein.

203. The '426 Patent was filed on July 11, 2019, and was duly and legally issued by the U.S. Patent and Trademark Office on Oct. 5, 2021, entitled, “Streaming media multiplexing with a media proxy.”

204. Snap One is the owner by assignment of the '426 Patent. A copy of the '426 Patent is attached hereto as Exhibit C. Snap One holds all rights, title, and interest to the '426 Patent, including the sole and exclusive right to bring a claim for its infringement.

205. The '426 Patent is valid, enforceable, and was duly issued in full compliance with Title 35 of the United States Code.

206. The '426 Patent is generally directed to a “method for communicating from one device to a plurality of receiving devices.”

207. The '426 Patent teaches technology improvements relating to an intercom or paging system for a multi-room audio format that provides a way for one calling device to use a media multiplexer and media proxy for broadcast communications with several callee devices and then to switch to exclusive communication with just one callee device. The inventions described in the '426 Patent represent improvements in the art.

208. Josh.ai manufactures and sells, among other things, intercom communication devices for the home that can communicate with several other devices.

209. The “Josh User Communication Features” includes at least any of the Josh Core as defined herein, the Josh Nano, as depicted at <https://www.josh.ai/nano/>, the Josh Micro as depicted at <https://www.josh.ai/micro/>, and features including the intercom, announcement, and other communication capabilities.

210. On information and belief, Josh.ai imports, makes, uses, sells, and/or offers to sell Josh User Communication Features, as well as components (which further includes any accessories) including software associated with and for use with the Josh User Communication Features to provide an intercom or paging system for a multi-room audio format that provides a way for one calling device to use a media multiplexer and media proxy for broadcast communications with several callee devices and then to switch to exclusive communication with just one callee device.

211. Josh.ai is liable for its infringement of the '426 Patent pursuant to 35 U.S.C. §§ 271(a), (b), and (c).

212. The Josh User Communication Features is covered, either literally or under the doctrine of equivalents, by one or more claims of the '426 Patent. For example, at a minimum, the Josh User Communication Features connect to a home automation network in a manner that embodies one or more claimed inventions in the '426 Patent, including at least claim 1. An example of the intercom system for a home automation network with the ability to communicate with several devices and in a two-way intercom functionality is shown below:

To kick things off, [Josh.ai](#) has announced the company's first touchless home intercom feature. With homeowners communicating to their Josh.ai systems every day and adding more microphones in areas of their homes, it was a natural next step to empower families to communicate directly with each other in their living spaces. While Josh.ai users were already using one-way announcements to send a message to a single room or throughout a home, two-way intercom functionality via Josh Micro and Josh Nano expands the possibilities.

By adding intercom functionality, Josh.ai is increasing its value and enabling households to connect in new ways. This feature is now available to new and existing Josh.ai clients at no additional charge.

<https://www.cepro.com/control/josh-ai-announces-updates-enhancements-during-cedia-expo-2021/>

213. As shown above and described below, the Josh User Communication Features and associated components/software meets the limitations of claim 1. For example, the Josh User Communication Features and associated components/software uses a plurality of receiving devices such as the Josh Nano and Josh Micro.

214. Josh User Communication Features and associated components/software includes receiving a request for a call and media from one device at a Josh Core or the Josh Nano. On

information and belief, the Josh Core then allocates or establishes communication channels for a plurality of receiving devices, such as the Josh Nano and Josh Micro.

215. On information and belief, the Josh User Communication Features closes communication channels for all receiving devices except one.

216. Josh.ai's conduct, as set forth herein, has directly infringed and continues to directly infringe one or more claims of the '426 Patent, including at least claim 1, by making, using, selling, offering for sale within the United States, and/or importing into the United States products, systems, and/or services, in violation of at least 35 U.S.C. §§ 271(a) and 281 *et seq.*

217. Additionally, Josh.ai has indirectly infringed and continues to indirectly infringe one or more of the claims of the '426 Patent, in violation of 35 U.S.C. § 271(b), by knowing and intentionally inducing infringement of the '623 Patent, literally and/or under the doctrine of equivalents. Josh.ai has and continues to actively induce their inducing its customers to sell and/or use the Josh User Communication Features in a manner directly infringing one or more claims of the '426 Patent, including at least claim 1.

218. On information and belief, (a) Josh.ai has had actual or constructive knowledge of the '426 Patent or, at least, has actual knowledge as of the filing of this Complaint, (b) Josh.ai intentionally causes, urges, or encourages users of the Josh User Communication Features to directly infringe one or more claims of the '426 Patent by, for example, promoting, advertising, and instructing customers and potential customers about the Josh User Communication Features and uses thereof, including infringing uses (c) Josh.ai knows (or should know) that its actions will induce customers/users of the Josh User Communication Features to directly infringe one or more claims the '426 Patent, and (d) customers/users of Josh User Communication Features

directly infringe one or more claims of the '426 Patent. In addition, Josh.ai has supplied and continues to supply the Josh User Communication Features to customers while knowing that use (and inducing such use) of the Josh User Communication Features will infringe one or more claims of the '426 Patent, and that Josh.ai's customers then directly infringe one or more claims of the '426 Patent by selling and/or using the Josh User Communication Features.

219. Additionally, Josh.ai has indirectly infringed and continue to indirectly infringe one or more of the claims of the '426 Patent, in violation of 35 U.S.C. § 271(c), by offering to sell or selling within the United States, or importing into the United States, components in connection with the Josh User Communication Features that contribute to the direct infringement of the '426 Patent by users of the Josh User Communication Features.

220. On information and belief, (a) Josh.ai has actual or constructive knowledge of the '426 Patent, or at least have knowledge at the time of this suit, (b) Josh.ai offers or sale, sells, or imports the Josh User Communication Features products that comprise one or more material components of the invention of the '426 Patent and are not staple articles of commerce suitable for substantial non-infringing use, (c) Josh.ai knows (or should know) that such component(s) are especially made or especially adapted for use in an infringement of the '426 Patent, and (d) customers/users of the Josh User Communication Features that comprise such material component(s) directly infringe one or more claims of the '426 Patent. For example, at a minimum, Josh.ai offers for sale, sells, and/or imports Josh User Communication Features that comprise software and/or hardware components adapted for use in one or more inventions claimed in the '426 Patent, and Josh.ai's customers then directly infringe one or more claims of

the '426 Patent by selling and/or using the Josh User Communication Features that comprise these components.

221. Josh.ai imports, makes, uses, sells, and/or offers to sale the Josh User Communication Features, despite an objectively high likelihood that their actions constitute infringement.

222. Snap One has satisfied the statutory requirements to collect pre-suit damages for infringement of each of the Snap One Patents, including, without limitation, any applicable provisions of 35 U.S.C. § 287, by, at least, providing notice via a website made available at <https://www.control4.com/legal/patents/>.

223. Josh.ai's acts of infringement have caused damage to Snap One, and Snap One is entitled to recover for the damages sustained because of Josh.ai's wrongful acts including lost profits but in no event less than a reasonable royalty in an amount to be determined including any further relief, such as compensatory or equitable, as deemed proper and just by the Court.

224. Snap One is also entitled to injunctive relief because Josh.ai's infringement of the '426 Patent will continue to damage Snap One's business, causing irreparable harm, for which there is no adequate remedy at law.

225. Josh.ai knew at least as early as the filing of this Complaint that, through its encouragement, the Josh User Communication Features would result in infringement of one or more claims of the '426 Patent.

226. Despite having actual knowledge, Josh.ai has continued to carry out infringing activities. Josh.ai has had actual knowledge, or at least constructive knowledge, of the '426

Patent. Josh.ai has known, or have been willfully blind, to the existence of the '426 Patent and associated infringement thereof.

227. Josh.ai has had notice and has willfully infringed and continue to willfully infringe the '426 Patent, entitling Snap One to increased damages under 35 U.S.C. § 284 and to attorney fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

228. By reason of the foregoing, Snap One is entitled to relief against Josh, pursuant to at least 35 U.S.C. §§ 283–85.

EIGHTH CLAIM FOR RELIEF

(Infringement of U.S. Patent No. 7,886,338 – Against Josh.ai)

229. Snap One incorporates by reference the allegations set forth in the preceding paragraphs as though fully set forth herein.

230. The '338 Patent was filed on Oct. 9, 2007, and was duly and legally issued by the U.S. Patent and Trademark Office on Feb. 8, 2011, entitled, “System and method for controlling access to local services without losing failover capibility [*sic*].”

231. Snap One is the owner by assignment of the '338 Patent. A copy of the '338 Patent is attached hereto as Exhibit D. Snap One holds all rights, title, and interest to the '338 Patent, including the sole and exclusive right to bring a claim for its infringement.

232. The '338 Patent is valid, enforceable, and was duly issued in full compliance with Title 35 of the United States Code.

233. The '338 Patent is generally directed to controlling devices in a home automation network, for example, to “restrict access to different services that are provided through automated, networked devices.”

234. The '338 Patent teaches technology improvements addressing and overcoming problems arising in centrally controlled homes, and, for example, teaches putting some networked devices (e.g., a speaker with a microphone) into an intermediate state (e.g., privacy mode) to restrict operation of the devices, and, for example, teaches a timer procedure to switch from fully operational to an intermediate state, thereby improving the art.

235. Josh.ai manufactures and sells, among other things, smart-home voice control devices that are integrated with a home automation network and products.

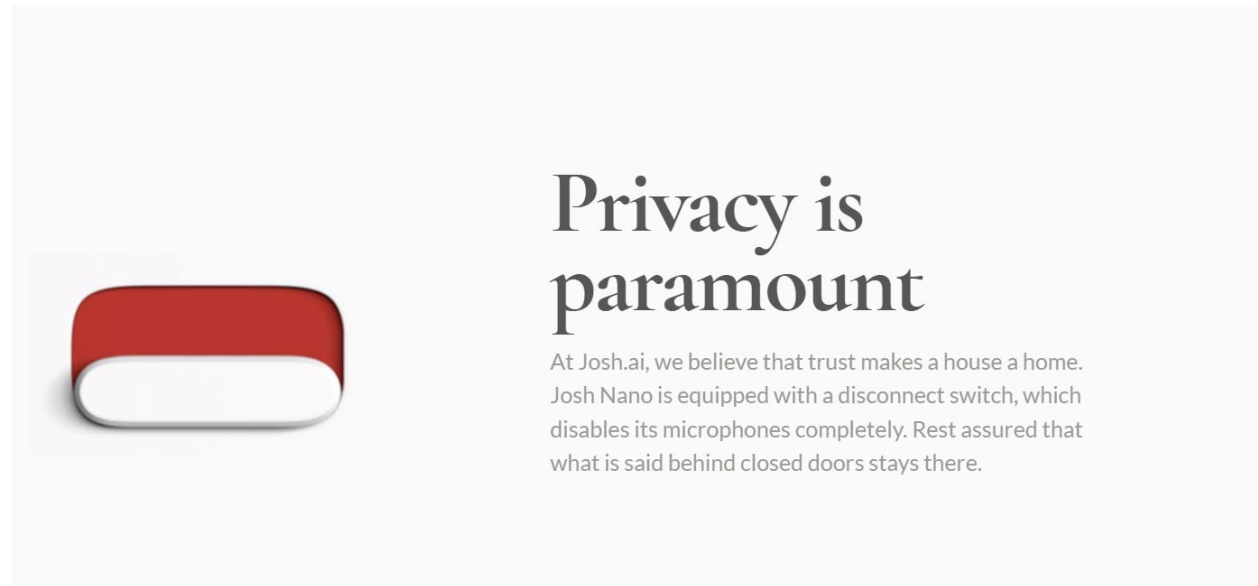
236. The "Josh Control Devices" includes at least the Josh Core, Josh Nano and Josh Micro.

237. On information and belief, Josh.ai imports, makes, uses, sells, and/or offers to sell Josh Control Devices, as well as components (which further includes any accessories) including software associated with and for use with the Josh Control Devices to provide access and integration to home automation networks that allows for some networked devices (i.e., a speaker with a microphone) to be put into an intermediate state (e.g., privacy mode) to restrict operation of the devices, and a timer procedure to switch from fully operational to an intermediate state.

238. Josh.ai is liable for its infringement of the '338 Patent pursuant to 35 U.S.C. §§ 271(a), (b), and (c).

239. The Josh Control Devices are covered, either literally or under the doctrine of equivalents, by one or more claims of the '338 Patent. For example, at a minimum, the Josh Control Devices connect to a home automation network in a manner that embodies one or more claimed inventions in the '338 Patent, including at least claim 1. Some examples of the

networked devices to be put in an offline state to restrict operation of the devices, and a timer procedure to switch from online to offline state. Is shown below:



240. As shown above and described below, the Josh Control Devices meet the limitations of claim 1. For example, the Josh Control Devices and associated components/software includes setting a state of a networked device to fully operational state.

241. The Josh Control Devices and associated components/software includes sending signaling to the Josh Core requesting changing the state of the device, such as, for example, but not limited to, signaling for putting the device into privacy mode and, on information and belief, timer procedures to switch from fully operational to an intermediate state, such as the privacy mode.

242. Josh.ai's conduct, as set forth herein, has directly infringed and continues to directly infringe one or more claims of the '338 Patent, including at least claim 1, by making, using, selling, offering for sale within the United States, and/or importing into the United States products, systems, and/or services, in violation of at least 35 U.S.C. §§ 271(a) and 281 *et seq.*

243. Additionally, Josh.ai has indirectly infringed and continues to indirectly infringe one or more of the claims of the '338 Patent, in violation of 35 U.S.C. § 271(b), by knowing and intentionally inducing infringement of the '338 Patent, literally and/or under the doctrine of equivalents. Josh.ai has and continues to actively induce its customers to sell and/or use the Josh Control Devices in a manner directly infringing one or more claims of the '338 Patent, including at least claim 1.

244. On information and belief, (a) Josh.ai has had actual or constructive knowledge of the '338 Patent, or at least have actual knowledge as of the filing of this Complaint, (b) Josh.ai intentionally causes, urges, or encourages users of the Josh Control Devices to directly infringe one or more claims of the '338 Patent, for example, by promoting, advertising, and instructing customers and potential customers about the Josh System and uses thereof, including infringing uses (c) Josh.ai knows (or should know) that their actions will induce customers/users of the Josh Control Devices to directly infringe one or more claims the '338 Patent, and (d) customers/users of Josh Control Devices directly infringe one or more claims of the '338 Patent. In addition, Josh.ai has supplied and continue to supply the Josh Control Devices to customers while knowing that use (and inducing such use) of the Josh Control Devices will infringe one or more claims of the '338 Patent, and that Josh.ai's customers then directly infringe one or more claims of the '338 Patent by selling and/or using the Josh Control Devices.

245. Additionally, Josh.ai has indirectly infringed and continues to indirectly infringe one or more of the claims of the '338 Patent, in violation of 35 U.S.C. § 271(c), by offering to sell or selling within the United States, or importing into the United States, components in

connection with the Josh Control Devices that contribute to the direct infringement of the '338 Patent by users of the Josh Control Devices.

246. On information and belief, (a) Josh.ai has actual or constructive knowledge of the '338 Patent, or at least has knowledge at the time of this suit, (b) Josh.ai offers or sale, sell, or import the Josh Control Devices products that comprise one or more material components of the invention of the '338 Patent and are not staple articles of commerce suitable for substantial non-infringing use, (c) Josh.ai knows (or should know) that such component(s) are especially made or especially adapted for use in an infringement of the '338 Patent, and (d) customers/users of the Josh Control Devices comprising such material component(s) directly infringe one or more claims of the '338 Patent. For example, at a minimum, Josh.ai offers for sale, sell, and/or import Josh Control Devices that comprise software and/or hardware components adapted for use in one or more inventions claimed in the '338 Patent, and Josh.ai's customers then directly infringe one or more claims of the '338 Patent by selling and/or using the Josh Control Devices that comprise these components.

247. Josh.ai imports, makes, uses, sells, and/or offers to sale the Josh Control Devices, despite an objectively high likelihood that their actions constitute infringement.

248. Snap One has satisfied the statutory requirements to collect pre-suit damages for infringement of each of the Snap One Patents, including, without limitation, any applicable provisions of 35 U.S.C. § 287, by, at least, providing notice via a website made available at <https://www.control4.com/legal/patents/>.

249. Josh.ai's acts of infringement have caused damage to Snap One, and Snap One is entitled to recover the damages sustained because of Josh.ai's wrongful acts including lost profits

but in no event less than a reasonable royalty in an amount to be determined including any further relief, such as compensatory or equitable, as deemed proper and just by the Court.

250. Snap One is also entitled to injunctive relief because Josh.ai's infringement of the '338 Patent will continue to damage Snap One's business, causing irreparable harm, for which there is no adequate remedy at law.

251. Josh.ai knew at least as early as the filing of this Complaint that, through its encouragement, the Josh Control Devices would result in infringement of one or more claims of the '338 Patent.

252. Despite having actual knowledge, Josh.ai has continued to carry out infringing activities. Josh.ai has had actual knowledge, or at least constructive knowledge, of the '338 Patent. Josh.ai has known, or have been willfully blind, to the existence of the '338 Patent and associated infringement thereof.

253. Josh.ai has had notice and have willfully infringed and continue to willfully infringe the '338 Patent, entitling Snap One to increased damages under 35 U.S.C. § 284 and to attorney fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

254. By reason of the foregoing, Snap One is entitled to relief against Josh, pursuant to at least 35 U.S.C. §§ 283–85.

NINTH CLAIM FOR RELIEF

(Infringement of U.S. Patent No. 10,756,984 – Against Josh.ai)

255. Snap One incorporates by reference the allegations set forth in the preceding paragraphs as though fully set forth herein.

256. The '984 Patent was filed on June 26, 2015, and was duly and legally issued by the U.S. Patent and Trademark Office on Aug. 25, 2020, entitled, "Method and apparatus for creating and managing network device port VLAN configurations."

257. Snap One is the owner by assignment of the '984 Patent. A copy of the '984 Patent is attached hereto as Exhibit E. Snap One holds all rights, title, and interest to the '984 Patent, including the sole and exclusive right to bring a claim for its infringement.

258. The '984 Patent is valid, enforceable, was duly and legally issued by the U.S. Patent and Trademark Office in full compliance with Title 35 of the United States Code.

259. The '984 Patent is generally directed to "a user interface for creating and managing network device port VLAN configurations."

260. The '984 Patent teaches technology improvements for a network, including a client, connected to a network device (e.g., a switch) having a plurality of ports, for configuring different ports of the network device to have different VLAN assignments using a graphical user interface provided by the client, thereby improving the art.

261. Josh.ai manufactures and sells, among other things, smart-home devices that are integrated with a home automation network and products.

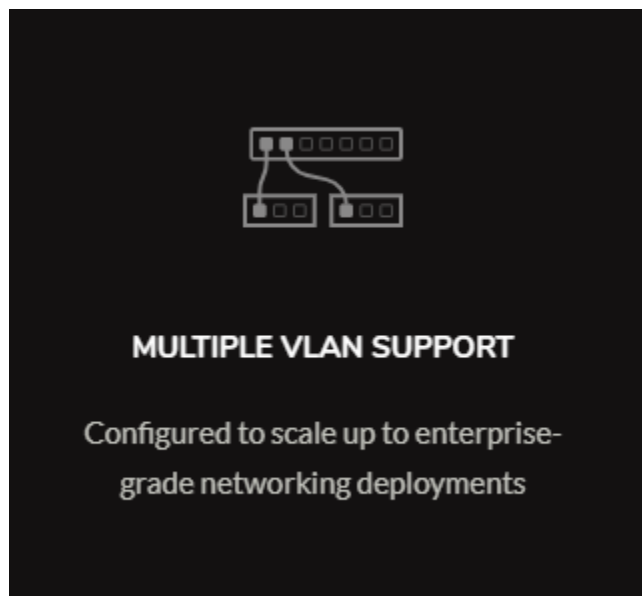
262. The "Josh Core" includes at least the Josh Core device as depicted at <https://www.josh.ai/core/>.

263. On information and belief, Josh.ai imports, makes, uses, sells, and/or offers to sell, as well as components (which further includes any accessories) including software associated with and for use with the Josh Core to provide a client, connected to a network device

(e.g., a switch) having a plurality of ports, for configuring different ports of the network device to have different VLAN assignments using a graphical user interface provided by the client.

264. Josh.ai is liable for its infringement of the '984 Patent pursuant to 35 U.S.C. §§ 271(a), (b), and (c).

265. The Josh Core is covered, either literally or under the doctrine of equivalents, by one or more claims of the '984 Patent. For example, at a minimum, the Josh Core connects to a home automation network in a manner that embodies one or more claimed inventions in the '984 Patent, including at least claim 1. Some examples of the system configured for multiple VLAN support including a graphical user interface for a plurality of ports is shown below:



<https://www.josh.ai/core/>

266. As shown above and described below, the Josh Core meets the limitations of claim 1.

267. For example, on information and belief, the Josh Core and associated components/software includes a network device that is configured with a graphical user interface

including a plurality of ports and a configuration interface enabling the user to make VLAN assignment for each of a plurality of ports, for example, that are connected to Josh Micros located in different rooms.

268. Josh.ai's conduct, as set forth herein, directly infringed and continues to directly infringe one or more claims of the '984 Patent, including at least claim 1, by making, using, selling, offering for sale within the United States, and/or importing into the United States products, systems, and/or services, in violation of at least 35 U.S.C. §§ 271(a) and 281 *et seq.*

269. Additionally, Josh.ai has indirectly infringed and continue to indirectly infringe one or more of the claims of the '984 Patent, in violation of 35 U.S.C. § 271(b), by knowing and intentionally inducing infringement of the '984 Patent, literally and/or under the doctrine of equivalents. Josh.ai has and continues to actively induce its customers to sell and/or use the Josh Core in a manner directly infringing one or more claims of the '984 Patent, including at least claim 1.

270. On information and belief, (a) Josh.ai has had actual or constructive knowledge of the '984 Patent, or at least have actual knowledge as of filing of this Complaint, (b) Josh.ai intentionally causes, urges, or encourages users of the Josh Core to directly infringe one or more claims of the '984 Patent by, for example, promoting, advertising, and instructing customers and potential customers about the Josh Core and uses thereof, including infringing uses, (c) Josh.ai knows (or should know) that its actions will induce customers/users of the Josh Core to directly infringe one or more claims the '984 Patent, and (d) customers/users of Josh Core directly infringe one or more claims of the '984 Patent. In addition, Josh.ai has supplied and continue to supply the Josh Core to customers while knowing that use (and inducing such use) of the Josh

Core will infringe one or more claims of the '984 Patent, and that Josh.ai's customers then directly infringe one or more claims of the '984 Patent by selling and/or using the Josh Core.

271. Additionally, Josh.ai has indirectly infringed and continue to indirectly infringe one or more of the claims of the '984 Patent, in violation of 35 U.S.C. § 271(c), by offering to sell or selling within the United States, or importing into the United States, components in connection with the Josh Core that contribute to the direct infringement of the '984 Patent by users of the Josh Core.

272. On information and belief, (a) Josh.ai has actual or constructive knowledge of the '984 Patent, or at least have knowledge at the time of this suit, (b) Josh.ai offers for sale, sells, or imports the Josh Core products that comprise one or more material components of the invention of the '984 Patent and are not staple articles of commerce suitable for substantial non-infringing use, (c) Josh.ai knows (or should know) that such component(s) are especially made or especially adapted for use in an infringement of the '984 Patent, and (d) customers/users of the Josh Core comprising such material component(s) directly infringe one or more claims of the '984 Patent. For example, at a minimum, Josh.ai offers for sale, sells, and/or imports Josh Core that comprise software and/or hardware components adapted for use in one or more inventions claimed in the '984 Patent, and Josh.ai's customers then directly infringe one or more claims of the '984 Patent by selling and/or using the Josh Core that comprise these components.

273. Josh.ai imports, makes, uses, sells, and/or offers to sale the Josh Core, despite an objectively high likelihood that their actions constitute infringement.

274. Snap One has satisfied the statutory requirements to collect pre-suit damages for infringement of each of the Snap One Patents, including, without limitation, any applicable

provisions of 35 U.S.C. § 287, by, at least, providing notice via a website made available at <https://www.control4.com/legal/patents/>.

275. Josh.ai's acts of infringement have caused damage to Snap One, and Snap One is entitled to recover the damages sustained because of Josh.ai's wrongful acts including lost profits but in no event less than a reasonable royalty in an amount to be determined including any further relief, such as compensatory or equitable, as deemed proper and just by the Court.

276. Snap One is also entitled to injunctive relief because Josh.ai's infringement of the '984 Patent will continue to damage Snap One's business, causing irreparable harm, for which there is no adequate remedy at law.

277. Josh.ai knew at least as early as the filing of this Complaint that, through its encouragement, the Josh Core would result in infringement of one or more claims of the '984 Patent.

278. Despite having actual knowledge, Josh.ai has continued to carry out infringing activities. Josh.ai has had actual knowledge, or at least constructive knowledge, of the '984 Patent. Josh.ai has known, or have been willfully blind, to the existence of the '984 Patent and associated infringement thereof.

279. Josh.ai has had notice and have willfully infringed and continue to willfully infringe the '984 Patent, entitling Snap One to increased damages under 35 U.S.C. § 284 and to attorney fees and costs incurred in prosecuting this action under 35 U.S.C. § 285.

280. By reason of the foregoing, Snap One is entitled to relief against Josh.ai pursuant to at least 35 U.S.C. §§ 283–85.

TENTH CLAIM FOR RELIEF

(Breach of Contract (Distributor Agreement) – Against Josh.ai)

281. Snap One incorporates by reference the allegations set forth in the preceding paragraphs as though set forth fully herein.

282. The Distributor Agreement is a valid and enforceable contract between Snap One and Josh.ai.

283. Snap One performed its obligations under the Distributor Agreement by acting in accordance with the terms set forth therein.

284. The Distributor Agreement granted Snap One the exclusive right to sell certain Josh.ai products to Snap One Control4 Dealers in North America and obligated Josh.ai to refer all Control4 Dealers in North America to purchase Josh.ai products through Snap One.

285. Josh.ai materially breached the Distributor Agreement by, among other things:

- a. Failing to refer all Control4 Dealers that desired to purchase Josh.ai products to Snap One; and
- b. Continuing to sell its products to new and existing Control4 Dealers at prices that were lower than Snap One (as a reseller of the products) could offer.

286. As a direct and proximate result of Josh.ai's breaches of the Distributor Agreement, Snap One has suffered, and continues to suffer, damages in an amount to be determined at trial but not less than the jurisdictional amount.

ELEVENTH CLAIM FOR RELIEF

(Breach of Implied Covenant of Good Faith and Fair Dealing – Against Josh.ai)

287. Snap One incorporates by reference the allegations set forth in the preceding paragraphs as though set forth fully herein.

288. The Distributor Agreement is a valid and enforceable contract between Snap One and Josh.ai.

289. Snap One performed its obligations under the Distributor Agreement by acting in accordance with the terms set forth therein.

290. Snap One and Josh.ai are subject to the implied covenant of good faith and fair dealing that adheres to the Distributor Agreement.

291. As described above, Josh.ai has breached the implied covenant of good faith and fair dealing by working to undermine the Distributor Agreement and eliminate Snap One's right to receive the full benefits of the contract.

292. Josh.ai breached the implied covenant of good faith and fair dealing by, among other things:

- a. Despite agreeing with the projected profit margins in the Distributor Agreement and (on information and belief) understanding Snap One's reliance on those margins, Josh.ai sold its products to Control4 Dealers at prices well below the suggested reseller prices, upon which the margins were based;
- b. On information and belief, Josh.ai used these lower prices to persuade Control4 Dealers to purchase Josh.ai products directly from Josh.ai instead of through Snap One; and
- c. Josh.ai relaxed the requirements for Dealers to make direct purchases from Josh.ai and employed promotional pricing that widened the gap between the prices being offered by Snap One and those being offered by Josh.

293. As a direct and proximate result of Josh.ai's breaches of the Distributor Agreement's implied covenant of good faith and fair dealing, Snap One has suffered, and continues to suffer, damages in an amount to be determined at trial.

JURY DEMAND

Snap One demands a jury trial on all issues triable by right of jury.

PRAYER FOR RELIEF

WHEREFORE, Snap One prays for judgment against Defendants as follows:

- A. On the First Claim for Relief:
 - 1. For general compensatory damages in an amount to be determined at trial.
- B. On the Second Claim for Relief:
 - 1. For general compensatory, statutory, and exemplary damages in an amount to be determined at trial;
 - 2. For an injunction prohibiting Defendants from further misappropriating Snap One's trade secrets; and
 - 3. For an award of attorney fees and costs.
- C. On the Third Claim for Relief:
 - 1. For general compensatory, statutory, and exemplary damages in an amount to be determined at trial;
 - 2. For an injunction prohibiting Defendants from further misappropriating Snap One's trade secrets; and
 - 3. For an award of attorney fees and costs.
- D. On the Fourth Claim for Relief:

1. For general compensatory and statutory damages in an amount to be determined at trial;
2. For an injunction prohibiting Defendants from further infringing on Snap One's trade dress related to the Neeo Remote;
3. For an injunction requiring Defendants to destroy all items infringing on Snap One's trade dress related to the Neeo Remote; and
4. For an award of attorney fees and costs

E. On the Fifth Claim for Relief:

1. For general compensatory and statutory damages including lost profits but in no event less than a reasonable royalty in an amount to be determined at trial;
2. For a declaration that Defendants have infringed the '623 Patent;
3. For a declaration that Defendants' infringement of the '623 Patent has been willful and deliberate;
4. For an injunction prohibiting Defendants from further infringing of the '623 Patent; and
5. For an award of pre-judgment and post-judgment interest and without limitation under 35 U.S.C. § 287; and
6. For an award of attorney fees and costs;

F. On the Sixth Claim for Relief:

1. For general compensatory and statutory damages including lost profits but in no event less than a reasonable royalty in an amount to be determined at trial;
2. For a declaration that AVA has infringed the '137 Patent;
3. For a declaration that AVA's infringement of the '137 Patent has been willful and deliberate;
4. For an injunction prohibiting AVA from further infringing the '137 Patent;
5. For an award of pre-judgment and post-judgment interest and without limitation under 35 U.S.C. § 287; and
6. For an award of attorney fees and costs.

G. On the Seventh Claim for Relief:

1. For general compensatory and statutory damages including lost profits but in no event less than a reasonable royalty in an amount to be determined at trial;
2. For a declaration that Josh.ai infringed the '426 Patent;
3. For a declaration that Josh.ai's infringement of the '426 Patent has been willful and deliberate;
4. For an injunction prohibiting Josh.ai from further infringing the '426 Patent;
5. For an award of pre-judgment and post-judgment interest and without limitation under 35 U.S.C. § 287; and

6. For an award of attorney fees and costs.

H. On the Eighth Claim for Relief:

1. For general compensatory and statutory damages including lost profits but in no event less than a reasonable royalty in an amount to be determined at trial;
2. For a declaration that Josh.ai infringed the '338 Patent;
3. For a declaration that Josh.ai's infringement of the '338 Patent has been willful and deliberate;
4. For an injunction prohibiting Josh.ai from further infringing the '338 Patent;
5. For an award of pre-judgment and post-judgment interest and without limitation under 35 U.S.C. § 287; and
6. For an award of attorney fees and costs.

I. On the Ninth Claim for Relief:

1. For general compensatory and statutory damages including lost profits but in no event less than a reasonable royalty in an amount to be determined at trial;
2. For a declaration that Josh.ai have infringed the '984 Patent;
3. For a declaration that Josh'ai's infringement of the '984 Patent has been willful and deliberate;
4. For an injunction prohibiting Josh.ai from further infringing the '984 Patent; and

5. For an award of pre-judgment and post-judgment interest and without limitation under 35 U.S.C. § 287;
 6. For an award of attorney fees and costs.
- J. On the Tenth Claim for Relief:
1. For general compensatory damages in an amount to be determined at trial.
- K. On the Eleventh Claim for Relief:
1. For general compensatory damages in an amount to be determined at trial.
- L. On All Causes of Action:
1. For an award of attorneys' fees and costs pursuant to SDK Agreement or as allowed by applicable law;
 2. For pre- and post-judgment interest as allowed by applicable law;
 3. For such other and further relief as this Court may deem appropriate under the circumstances.

DATED: August 14, 2023

FOLEY & LARDNER LLP

/s/ David L. Mortensen

David L. Mortensen

Tanner B. Camp

Attorneys for Snap One LLC

Exhibit A

(12) **United States Patent**
Barnum et al.

(10) **Patent No.:** **US 10,877,623 B2**
 (45) **Date of Patent:** **Dec. 29, 2020**

(54) **DYNAMIC INTERFACE FOR REMOTE CONTROL OF A HOME AUTOMATION NETWORK**

(75) Inventors: **Wally Barnum**, Salt Lake City, UT (US); **Joe Doubek**, Cottonwood Heights, UT (US)

(73) Assignee: **WirePath Home Systems, LLC**, Salt Lake City, UT (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 1651 days.

(21) Appl. No.: **12/139,356**

(22) Filed: **Jun. 13, 2008**

(65) **Prior Publication Data**
 US 2008/0313566 A1 Dec. 18, 2008

Related U.S. Application Data

(60) Provisional application No. 60/944,753, filed on Jun. 18, 2007.

(51) **Int. Cl.**
G06F 3/0482 (2013.01)
H04L 12/28 (2006.01)

(52) **U.S. Cl.**
 CPC **G06F 3/0482** (2013.01); **H04L 12/281** (2013.01); **H04L 12/2816** (2013.01); **G08C 2201/20** (2013.01); **G08C 2201/30** (2013.01)

(58) **Field of Classification Search**
 CPC G09G 5/14; H04N 5/44591; G06F 3/0481; G06F 3/0483; G06F 9/4443; H04L 12/281; H04L 12/2816; G08C 2201/20; G08C 2201/30
 USPC 715/810, 764, 709
 See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,878,377 A	4/1975	Brunone
3,909,826 A	9/1975	Schildmeier et al.
4,040,060 A	8/1977	Kaloi
4,068,289 A	1/1978	Ferrigno
4,138,684 A	2/1979	Kerr
4,163,218 A	7/1979	Wu
4,186,381 A	1/1980	Fleischer et al.
4,222,564 A	9/1980	Allen et al.
4,322,842 A	3/1982	Martinez
4,334,171 A	6/1982	Parman et al.
4,393,277 A	7/1983	Besen et al.
4,418,333 A	11/1983	Schwarzbach et al.
4,511,887 A	4/1985	Fiore

(Continued)

FOREIGN PATENT DOCUMENTS

EP	0 513 443	11/1992
EP	0 513 688	11/1992

OTHER PUBLICATIONS

Deposition of Bill Nuffer, CPP, Apr. 30, 2005, pp. 1-49.
 Anova Control Center , Operators Manual, pp. 1-50.

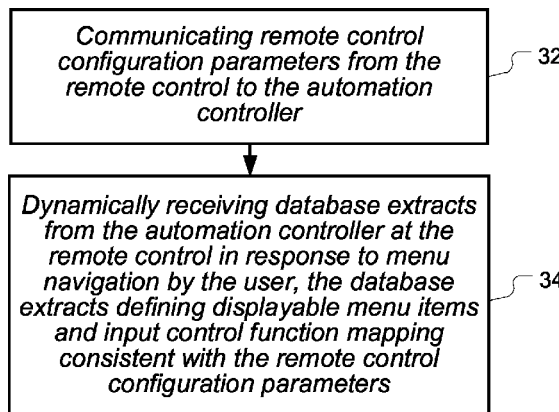
Primary Examiner — Arpan P. Savla
Assistant Examiner — Erik V Stitt
 (74) *Attorney, Agent, or Firm* — Austin Rapp

(57) **ABSTRACT**

A home automation network system having an automation controller and a remote control is described. The automation controller can dynamically download extracts from a menu database to the remote control as a user navigates through the menus. Downloading can be based on remote control configuration parameters communicated to the automation controller. Menus can be selected based on display limitations and input control types available on the remote control.

26 Claims, 8 Drawing Sheets

30



US 10,877,623 B2

Page 2

(56)

References Cited

U.S. PATENT DOCUMENTS

4,524,288	A	6/1985	Schimmelpennink et al.	5,455,464	A	10/1995	Gosling	
4,527,247	A	7/1985	Kaiser et al.	5,495,406	A	2/1996	Kushiro et al.	
4,636,791	A	1/1987	Burke et al.	5,530,322	A	6/1996	Ferrence et al.	
4,644,320	A	2/1987	Carr et al.	5,544,036	A	8/1996	Brown, Jr. et al.	
4,703,306	A	10/1987	Barrit	5,565,855	A	10/1996	Knibbe	
4,706,274	A	11/1987	Baker et al.	5,570,085	A	10/1996	Bertsch	
4,755,792	A	7/1988	Pezzolo et al.	5,572,438	A	11/1996	Ehlers et al.	
4,764,981	A	8/1988	Miyahara et al.	5,574,748	A	11/1996	Vander Mey et al.	
4,768,218	A	8/1988	Yorita	5,598,039	A	1/1997	Weber	
4,823,069	A	4/1989	Callahan et al.	5,736,965	A	4/1998	Mosebrook et al.	
4,825,209	A	4/1989	Sasaki et al.	5,778,372	A *	7/1998	Cordell et al.	
4,829,554	A	5/1989	Barnes et al.	5,793,300	A	8/1998	Suman et al.	
4,864,588	A	9/1989	Simpson et al.	5,798,581	A	8/1998	Keagy et al.	
4,873,711	A	10/1989	Roberts et al.	5,905,442	A	5/1999	Mosebrook et al.	
4,881,259	A	11/1989	Scordato	5,982,103	A	11/1999	Mosebrook et al.	
4,889,999	A	12/1989	Rowen	6,127,941	A	10/2000	Van Ryzin	
4,896,370	A	1/1990	Kasparian et al.	6,253,114	B1 *	6/2001	Takahara	G11B 27/002 700/17
4,908,604	A	3/1990	Jacob	6,341,316	B1 *	1/2002	Kloba et al.	709/248
4,918,432	A	4/1990	Pauley et al.	6,661,437	B1	12/2003	Miller et al.	
4,918,717	A	4/1990	Bissonnette et al.	6,687,487	B1	2/2004	Mosebrook et al.	
4,928,778	A	5/1990	Tin	6,791,467	B1	9/2004	Ben-Ze'ev	
4,939,792	A	7/1990	Urbish et al.	6,803,728	B2	10/2004	Balasubramaniam	
4,940,964	A	7/1990	Dao	6,983,424	B1 *	1/2006	Dutta	G06F 3/04817 715/790
4,977,529	A	12/1990	Gregg et al.	7,009,528	B2	3/2006	Griep	
4,996,703	A	2/1991	Gray	7,020,881	B2 *	3/2006	Takahashi	G06F 3/0481 348/E5.102
5,017,837	A	5/1991	Hanna et al.	7,032,176	B2	4/2006	Gordon et al.	
5,051,720	A	9/1991	Kittirutsunetorn	7,032,181	B1 *	4/2006	Farcasiu	H04M 1/274516 455/566
5,079,559	A	1/1992	Umetsu et al.	2001/0026291	A1	10/2001	Uchida	
5,086,385	A	2/1992	Launey et al.	2001/0030667	A1 *	10/2001	Kelts	345/854
5,089,974	A	2/1992	Demeyer et al.	2002/0171763	A1	11/2002	Stecyk et al.	
5,099,193	A	3/1992	Moseley et al.	2003/0048295	A1 *	3/2003	Lilleness et al.	345/738
5,109,222	A	4/1992	Welty	2003/0103088	A1 *	6/2003	Dresti et al.	345/835
5,146,153	A	9/1992	Luchaco et al.	2003/0169289	A1 *	9/2003	Holt et al.	345/735
5,173,855	A	12/1992	Nielsen et al.	2004/0230572	A1 *	11/2004	Omoigui	G06F 16/3338
5,191,265	A	3/1993	D'Aleo et al.	2005/0027539	A1 *	2/2005	Weber	G08C 17/00 704/275
5,218,344	A	6/1993	Ricketts	2005/0044164	A1 *	2/2005	O'Farrell et al.	709/213
5,218,628	A	6/1993	Ito	2005/0210057	A1 *	9/2005	Sprague	G06F 16/957
5,224,648	A	7/1993	Simon et al.	2006/0158438	A1 *	7/2006	Hunter	345/173
5,237,264	A	8/1993	Moseley et al.	2006/0161865	A1 *	7/2006	Scott et al.	715/810
5,321,736	A	6/1994	Beasley	2007/0186182	A1 *	8/2007	Schiller	715/781
5,348,078	A	9/1994	Dushane et al.	2013/0009900	A1 *	1/2013	Pryor	B60K 35/00 345/173
5,359,625	A	10/1994	Vander Mey et al.	2015/0067738	A1 *	3/2015	Lilleness	G06F 3/0488 725/58
5,382,947	A	1/1995	Thaler et al.					
5,386,209	A	1/1995	Thomas					
5,400,041	A	3/1995	Strickland					
5,400,246	A	3/1995	Wilson et al.					
5,426,439	A	6/1995	Grossman					

* cited by examiner

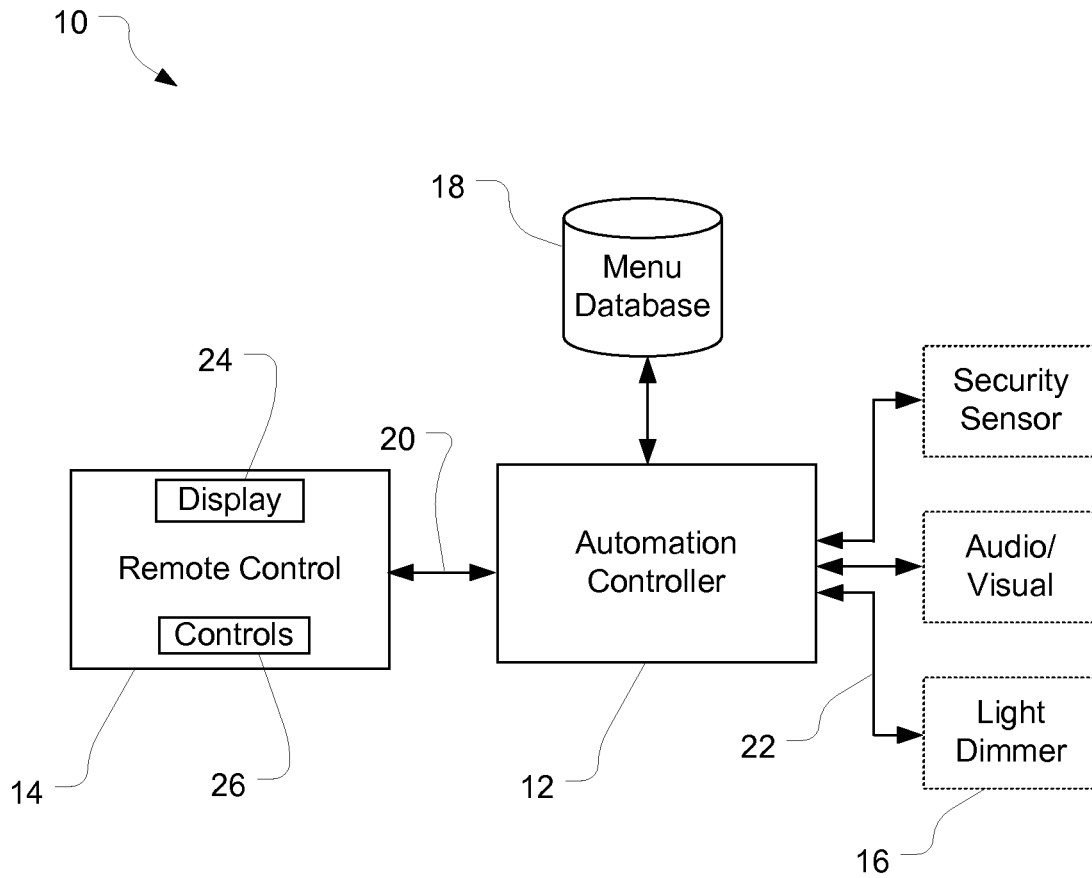


FIG. 1

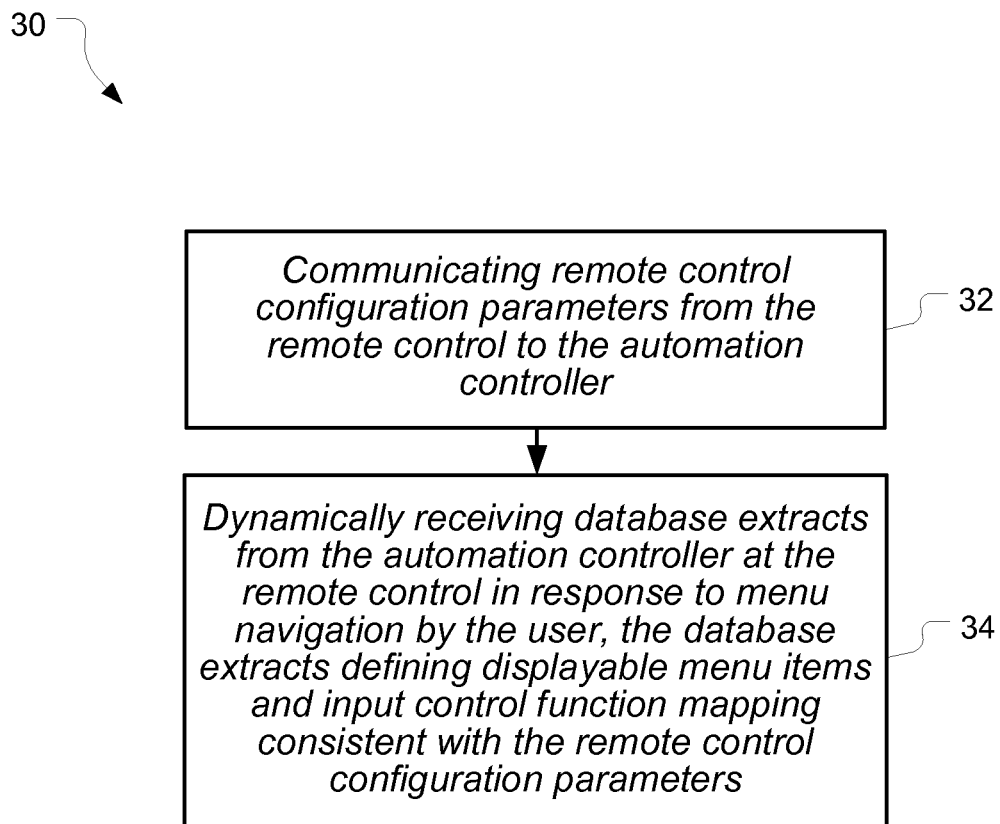


FIG. 2

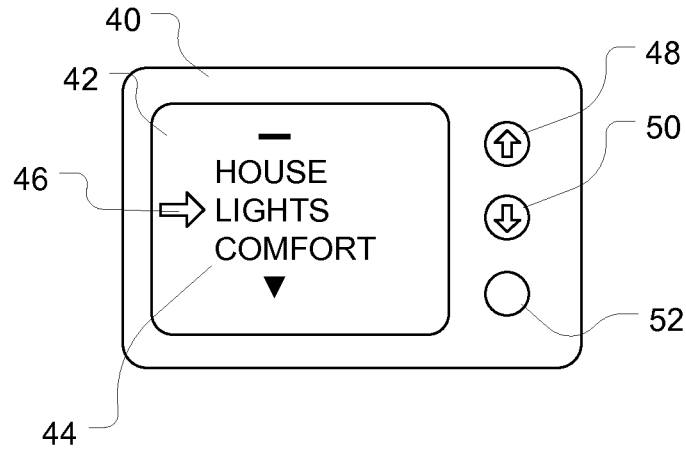


FIG. 3a

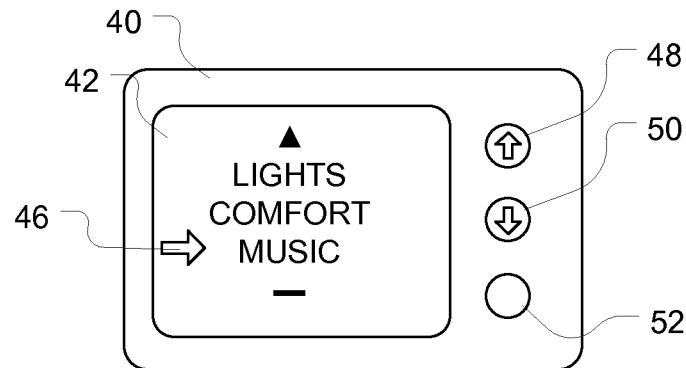


FIG. 3b

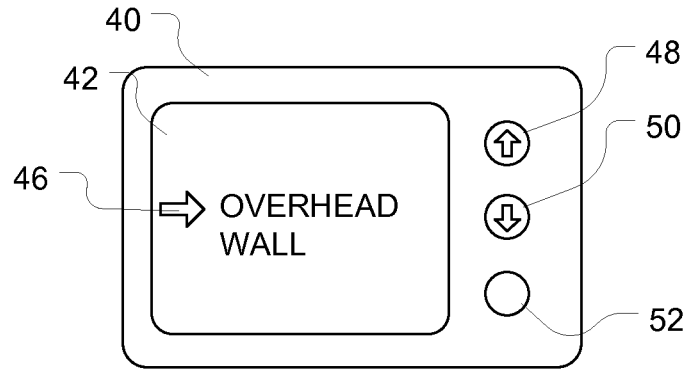


FIG. 3c

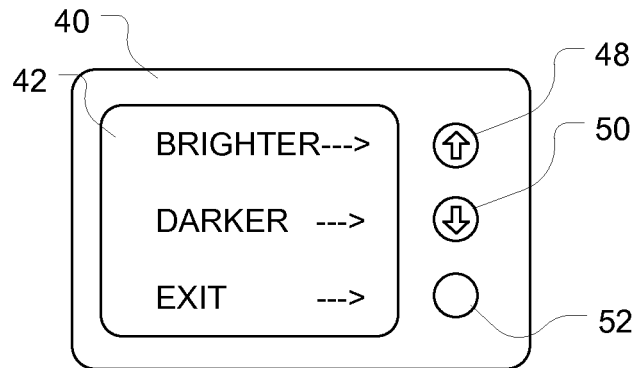


FIG. 3d

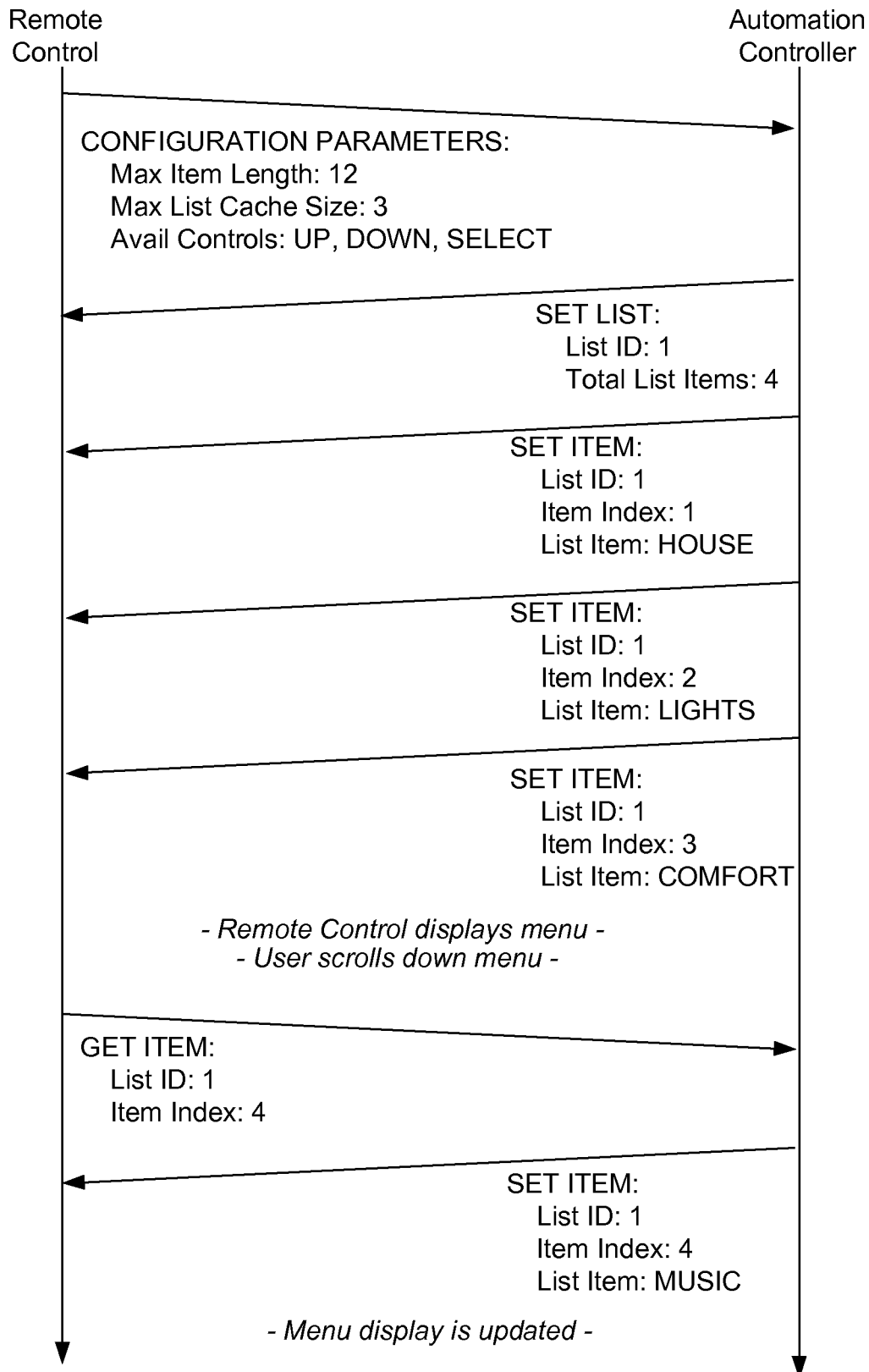


FIG. 4a

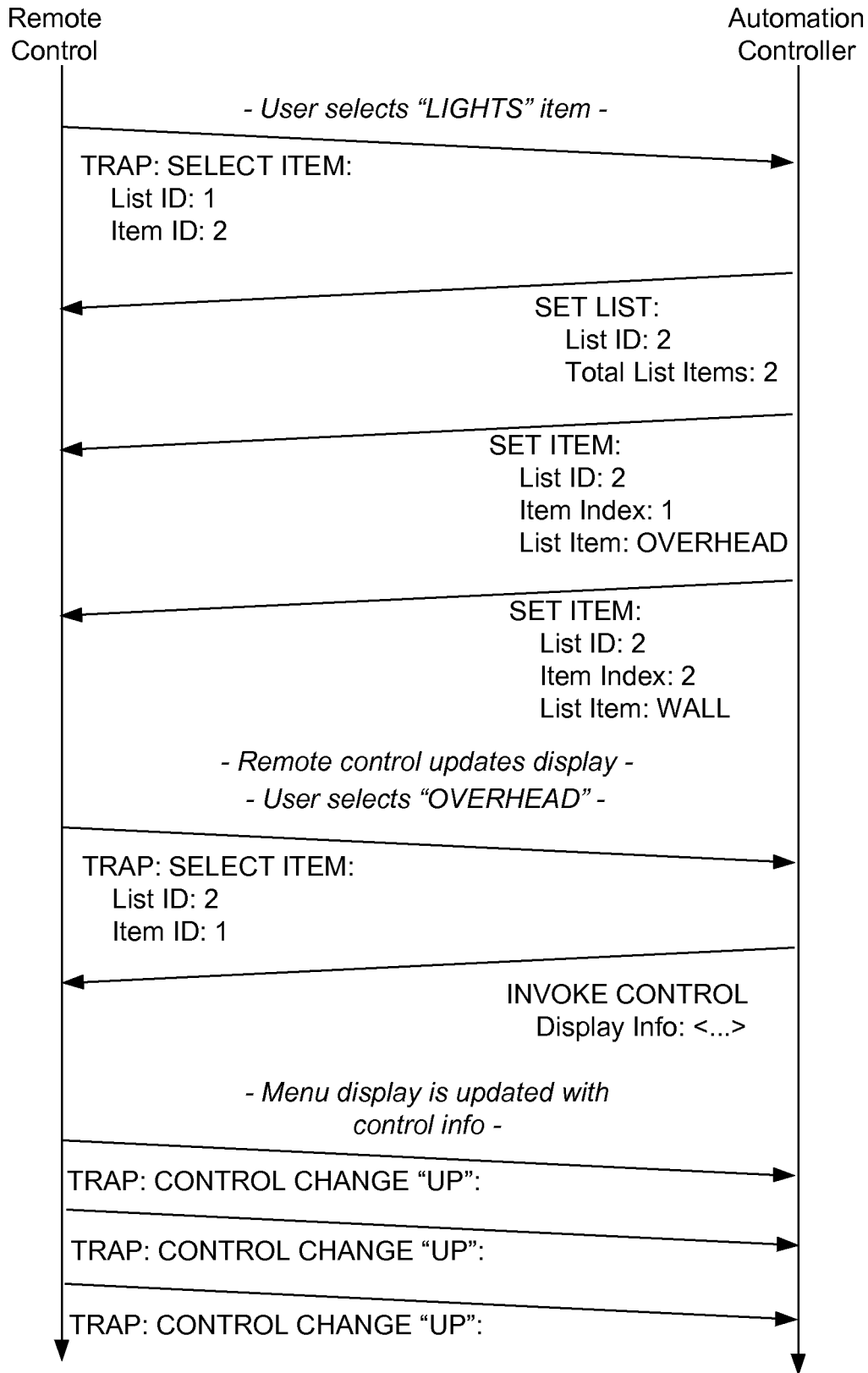


FIG. 4b

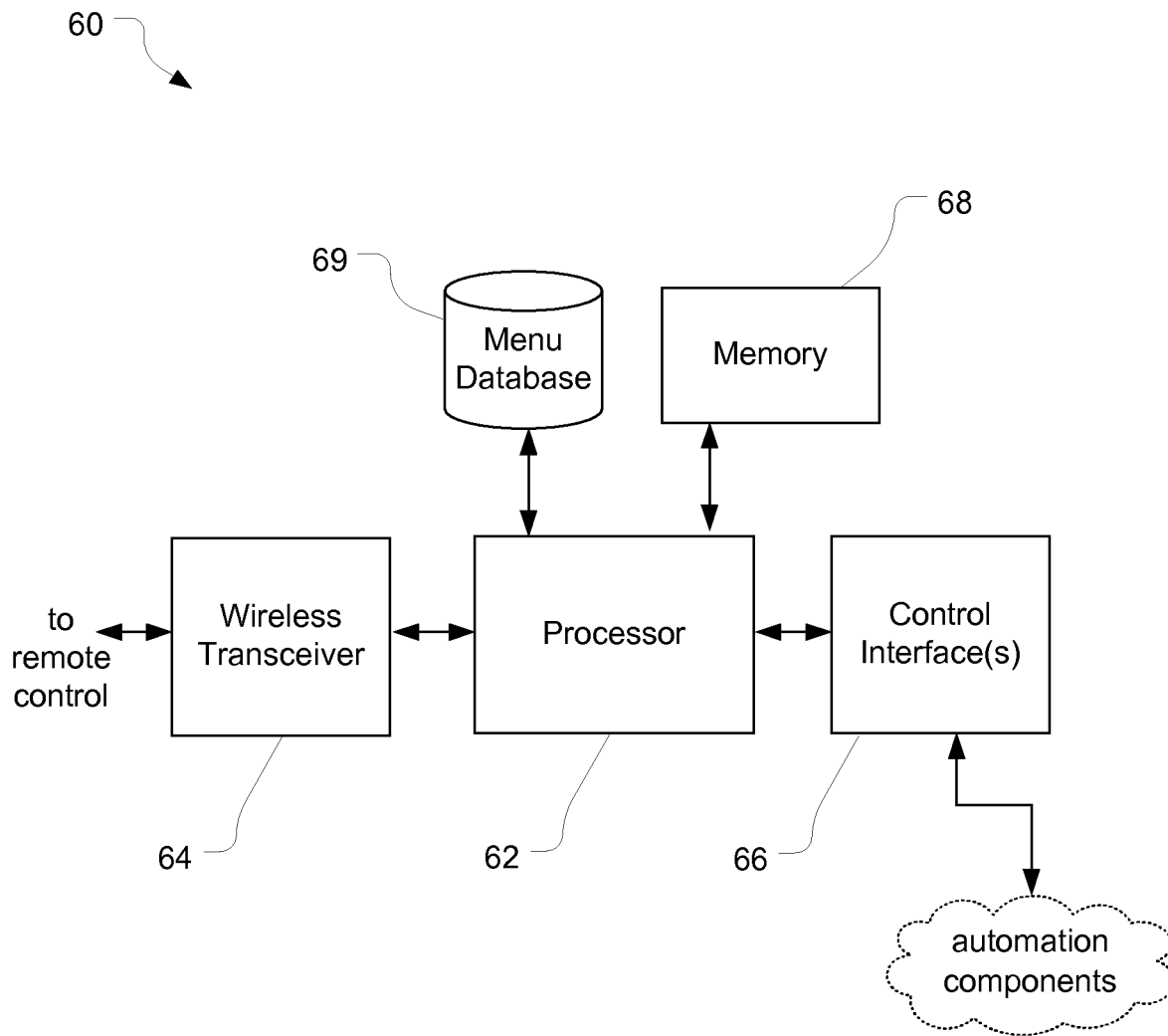


FIG. 5

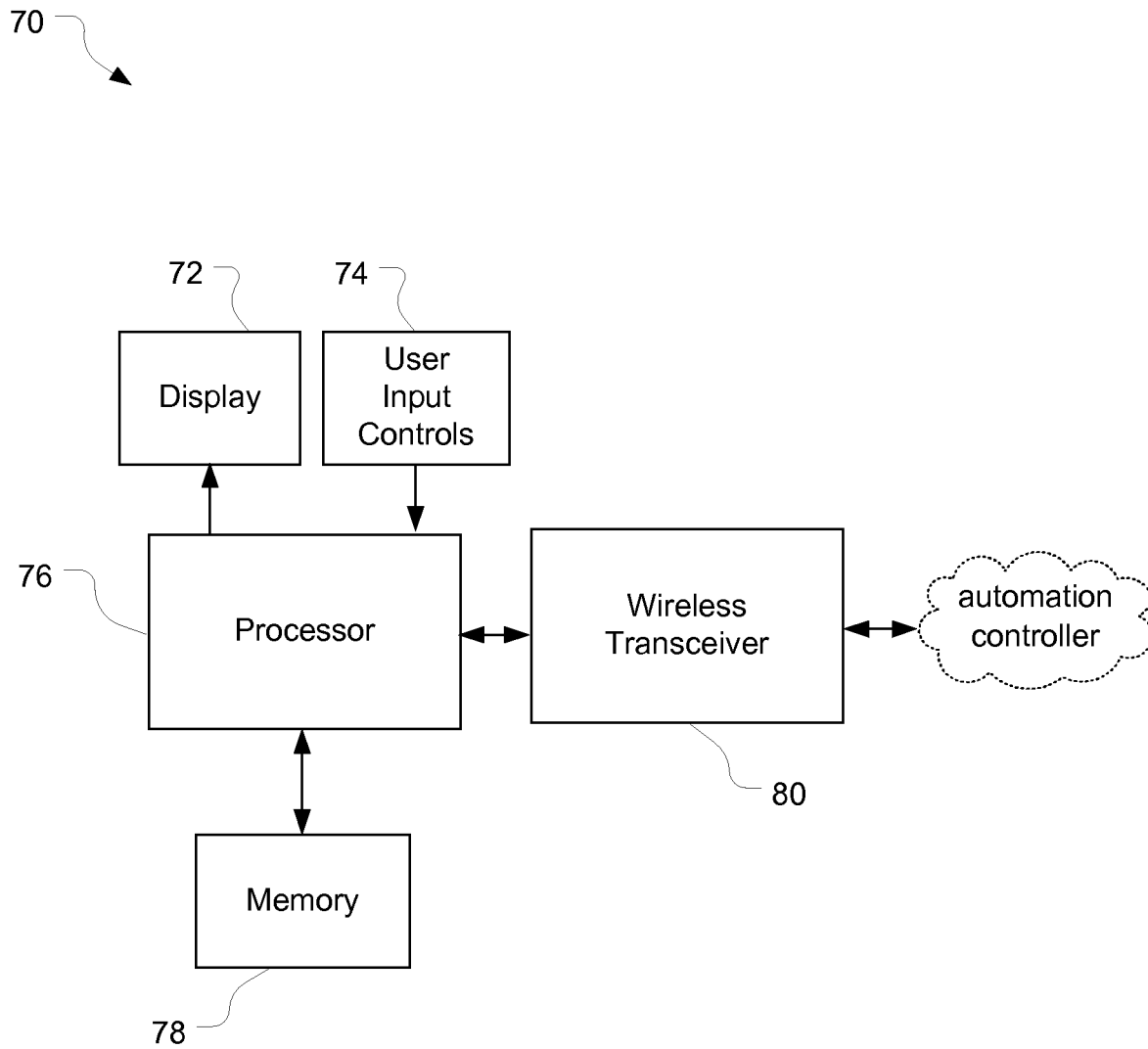


FIG. 6

US 10,877,623 B2

1

**DYNAMIC INTERFACE FOR REMOTE
CONTROL OF A HOME AUTOMATION
NETWORK**

CROSS-REFERENCE TO RELATED
APPLICATIONS AND CLAIM OF PRIORITY

Priority of U.S. Provisional patent application Ser. No. 60/944,753 filed on Jun. 18, 2007 is claimed.

BACKGROUND

Home automation networks have become increasingly popular. Home automation networks can incorporate diverse elements, such as security system components, environmental controls, lighting controls, and audio/visual equipment. While users enjoy the ability to control a wide variety of components, the necessary remote control devices can become very complex. With complexity comes cost and difficulties in upgrading and maintaining the controls.

For example, various so-called universal remote control devices are known in the art for audio/visual control, typically using infrared interfaces. These universal remote control devices typically include a large database of control codes for controlling audio/visual equipment such as DVD players, VCRs, televisions, stereo receivers, home theatre systems, and the like. The large database may be pre-programmed, making it difficult to update the universal remote control to accommodate new devices which become available.

Some universal remote controls can be programmed to learn new control codes. For example, the devices can be connected to the Internet through a computer in order to download new control codes for new devices. Other devices can be programmed with new codes using the original remote control provided with the device. Although devices can be updated to accommodate new system components, reprogramming the remote control is cumbersome for a user.

Most universal remote control devices are targeted at audio/visual control, and do not typically include the ability to control other equipment, such as environmental or lighting controls. The devices also typically have a number of buttons which are allocated to fixed functions, such as volume control, input selection, channel changing, and the like.

When a large number of devices are to be controlled by the remote control device, the remote control can become quite complex. Some remote control devices include a very large number of buttons, which can be visually undesirable as well as drive up costs for the packaging of the device.

For a complex home automation network, it is often desirable to provide multiple remote controls, some portable and some in fixed locations. For example, a remote control may be mounted at a wall location where it is primarily used for controlling lighting. Another remote control may be mounted near a door where it is primarily used for security system control. Using a universal type remote control at each location can be cost prohibitive. While less expensive, single purpose controls can be provided, they may not be able to provide all of the desired user functions.

SUMMARY

In one embodiment of the invention, a method of operating a remote control includes communicating remote control configuration parameters from the remote control to an automation controller. The method can also include dynami-

2

cally receiving database extracts from the automation controller at the remote control in response to menu navigation by the user. The database extracts can define user interface elements which are displayable on a display of the remote control, consistent with the remote control configuration parameters.

In another embodiment of the invention, a method of control in a home automation network includes communicating remote control capacity parameters from a remote control to an automation controller. Another step of the method is downloading menu items from the automation controller to the remote control, where the number and length of the menu items is selected by the automation controller based on the remote control capacity parameters. The method can also include displaying menu items on a display of the remote control and downloading additional menu items in response to user navigation commands.

Another embodiment of the invention includes a home automation network having an automation controller. The automation controller is coupled to a plurality of home automation network components. A menu database is accessible to the automation controller and defines user-navigable control menus for control of the home automation network. A remote control is in communication with the automation controller and can dynamically download portions of the menu database in response to a user navigating through the menu database using the remote control.

Additional embodiments of the invention include an automation controller having means for extracting and communicating menu items to a remote control in response to requests from the remote control, and a remote control having means for dynamically downloading a limited number of menu items of a limited length from an automation controller as a user navigates through a displayed menu.

BRIEF DESCRIPTION OF THE DRAWINGS

Additional features and advantages of the invention will be apparent from the detailed description which follows, taken in conjunction with the accompanying drawings, which together illustrate, by way of example, features of the invention; and, wherein:

FIG. 1 is a block diagram of a home automation network in accordance with an embodiment of the present invention;

FIG. 2 is a flow chart of the method of remote control in a home automation network in accordance with an embodiment of the present invention;

FIGS. 3a through 3d are illustrations of a remote control display in accordance with an embodiment of the present invention;

FIGS. 4a and 4b are protocol flow diagrams of message exchange between a remote control and an automation controller in accordance with an embodiment of the present invention;

FIG. 5 is a block diagram of a detailed implementation of an automation controller in accordance with an embodiment of the present invention; and

FIG. 6 is a block diagram of a detailed implementation of a remote control in accordance with an embodiment of the present invention.

Reference will now be made to the exemplary embodiments illustrated, and specific language will be used herein to describe the same. It will nevertheless be understood that no limitation of the scope of the invention is thereby intended.

DETAILED DESCRIPTION OF EXAMPLE EMBODIMENT(S)

FIG. 1 provides a block diagram of a home automation network in accordance with an embodiment of the present invention. The home automation network, shown generally at 10, includes an automation controller 12 and a remote control 14. The automation controller provides control of various home automation network components 16, including for example, lighting dimmers and switches, audio/visual equipment, climate control, and the like. A menu database 18, accessible to the automation controller, defines displayable user interface elements to allow user control of the home automation network. Displayable user interface elements can include, for example, displayable menu items, displayable virtual controls, and control function mapping, as described further herein. The menu database may, for example, be stored in memory in the automation controller as discussed further below.

The remote control 14 is in communication with the automation controller 12. The remote control is arranged to dynamically download limited portions of the menu database from the automation controller in response to a user navigating through the menu database using the remote control.

For example, the communication link 20 between the automation controller 12 and the remote control 14 can be a wireless communication link. The automation controller and the remote control can each include a wireless transceiver to form a wireless network as described further below. Communication 22 from the automation controller to some or all of the home automation network components 16 can also be via the wireless network.

The remote control 14 can request the limited portions of the menu database 18 from the controller based on which menu items will need to be displayed to a user on a display 24 located on the remote control. For example, the remote control may request menu items that are not currently being displayed, but which are adjacent to a displayed menu item and thus might be scrolled to by a user using controls 26 located on the remote control. The limited portions of the menu database can be based on memory or display limitations of the remote control.

Various protocols can be used to implement the downloading of limited portions of the menu database. For example, FIG. 2 illustrates a flow chart for a method of operating a remote control. The method, shown generally at 30, includes communicating 32 remote control configuration parameters from the remote control to the automation controller. For example, remote control configuration parameters can include remote control capacity limits, such as display capacity limits, display width limit (e.g. in pixels or text characters), display height limit (e.g. in pixels or text lines), or the like. Remote control configuration parameters can also include a list of available physical controls (e.g. buttons, sliders, knobs) on the remote or stored virtual controls (e.g. touch screen implemented buttons and sliders), and similar items.

Communicating these configuration parameters allows the automation controller to adapt the menu items communicated to the remote control to the capabilities of the remote control. For example, a given menu item may be displayable as a short text using abbreviations, a long text, or a graphical image. The central controller can select which version of a given menu item to provide to the remote control that is consistent with the remote control configuration parameters. For example, a remote control which has a 30 character wide

display might be provided with short text versions of the menu items, while a remote control which has a larger display might be provided a longer text version. In other words, the number of menu items, and a length of each menu can be selected by the automation controller based on the dynamically communicated remote control capacity limits.

The method 30 can include dynamically receiving 34 database extracts from the automation controller at the remote control in response to menu navigation by the user. For example, as a user navigates through the menu, the menu item list may be scrolled on the display or new lists displayed. Rather than requiring the remote control to include memory for all of the possible menu items as the user navigates the menu, the menu items can be dynamically downloaded from the automation controller to the remote control. For example, a remote control with 4 lines of menu display may have a memory large enough for 6 menu items: the 4 being displayed plus 1 menu item above and 1 menu item below the currently displayed items. This can help to provide for rapid display updates as the user scrolls through the menu, since while the menu display is being updated, new menu items can be received and stored into memory. As another example, a remote control having only enough memory for one displayed menu item, may receive menu items for immediate display. Newly received menu items may replace menu items currently in memory.

Downloaded menu items may also include various user control definitions for physical or virtual controls available on the remote control. For example, a user may control the volume of an audio component through the remote control. Upon selecting the component to be controlled and the parameter (volume) being adjusted, the remote control may activate a control specified by the menu database. Various controls may be present on the remote control, including physical controls such as sliders, push buttons, toggle switches, rotary knobs. Virtual controls can also be included, for example implemented using a touch sensitive screen to provide touch sliders, touch switches, touch knobs, and the like. Multiple controls may be present on the remote control. The remote control configuration parameters can include control type identifier(s) to identify the type(s) of control(s) available on the remote control.

Extracts from the database downloaded to the remote control may define function mapping for the input controls on the remote control. For example, when the user selects a volume adjustment, a slider control may be activated when the remote control has a slider control. If no slider control is available, push buttons may be mapped to “volume up” and “volume down” functions. These different options can be defined in the menu database, and selected based on the remote control configuration. Because the automation controller is aware of the remote control configuration parameters, function mapping of input controls can be made accordingly.

Operation of the method will be described in further detail with reference to FIGS. 3a through 3d, which illustrates a series of simulated display screens on a remote control 40 as a user navigates a set of exemplary menus and in conjunction with FIGS. 4a and 4b which illustrate a protocol flow diagram showing one possible implementation of message exchanges between the automation controller and the remote control used to dynamically download the menu information. In the protocol flow diagrams of FIGS. 4a and 4b, time progresses downward. The remote control is represented by a vertical line to the left of the diagram and the automation controller is represented by a vertical line to the right of the diagram. Message flow is shown by the slanted lines

US 10,877,623 B2

5

between the vertical lines, corresponding to messages between the remote control and automation controller.

FIG. 3a illustrates a first menu on the display 42, where the user is presented a top level selection-type menu 44. The menu information is obtained as shown in FIG. 4a. The protocol sequence begins by the remote control communicating configuration parameters to the automation controller using a CONFIGURATION PARAMETER message. These configuration parameters are used by the automation controller to determine the particular menu items and how many menu items to extract from the database and transmit to the remote control. The automation controller can inform the remote control about the menu list (e.g. how many items) using a SET LIST message, and can then populate a portion of the list on the remote control using SET ITEM messages. The number of items populated in the list may be less than the total list length, consistent with the memory limitations of the remote control. In this example, the top level menu has four items, of which the first three "HOUSE", "LIGHTS", and "COMFORT" are sent to the remote control. The remote control then displays the menu as shown in FIG. 3a.

The user can navigate the menu, for example by moving a cursor 46 up and down the displayed menu, or by scrolling the menu, using the up control 48 and down control 50. In other words, the up control and down control are function mapped to a menu navigation function. Various other ways of displaying and navigating a menu list are possible and can be used in embodiments of the present invention.

By moving down to the bottom of the list, the menu list 44 may scroll on the display 42 to show an additional menu item "MUSIC" as illustrated in FIG. 3b. This additional menu item may be automatically received by the remote controller from the automation controller in response to the user menu navigation. For example, downloading of new menu items can be initiated by the remote controller sending a request to the automation controller, and the automation controller responding by sending the menu item to the remote control. The bottom portion of FIG. 4a illustrates this exchange, where the remote control initiates transfer with a GET ITEM message, to which the automation controller responds with a SET ITEM message providing menu item 4, "COMFORT".

As a user scrolls through the menu, the remote control may then request additional menu items from the automation controller using the GET ITEM message. New menu list items may overwrite menu items already in memory. As the user scrolls through the menu, depending on how much cache memory is available in the remote control as compared to the menu list length, the remote control may make a number of requests for new menu items from the automation controller. While the above example is a simple illustration, it should be appreciated that this technique can be valuable when a large number of menu items are available. For example, when browsing a music collection many hundreds of titles may be available that greatly exceeds the limited memory capacity within the remote control. Hence, the remote control can download, store, and display a limited subset of the titles, downloading additional titles and updating the display as the user scrolls through the selections. Accordingly, the technique scales nicely, allowing menus having a wide range of lengths to be handled consistently.

Continuing the example, the user may scroll back to the menu item "LIGHTS" and select that menu item by pressing the blank control 52. In other words, the blank control 52 has been mapped to a menu selection function. For example, by

6

selecting "LIGHTS" the user may be directed to a second menu display while allows selection of various lights in the current room as illustrated in FIG. 3c. Again, these menu items may be received from the automation controller in response to a request by the remote control for menu items from the automation controller. This sequence is illustrated in the top portion of FIG. 4b. Upon user-selection of the LIGHTS item, a TRAP: SELECT ITEM message is sent to the automation controller and new menu selections are provided to the remote control by SET ITEM messages.

Upon selecting a particular light, e.g. "OVERHEAD" a control-type screen may be presented as illustrated in FIG. 3d. Here, the blank control 52 has been mapped to a control selection function. Upon the user-selection of the particular light being controlled, the TRAP:SELECT ITEM message is sent to the automation controller as shown in the center of FIG. 4b. The automation controller responds with an INVOKE CONTROL message which includes display information (represented by "<...") and not shown in detail) for how the display is to appear as shown in FIG. 3d.

The control screen indicates the functions being performed by the input controls, showing that depressing the up key 48 will increase brightness of the overhead lighting and pressing the down key 50 will decrease brightness, and the blank key 52 will exit the control menu. As the up key and down key are pressed, the remote control can communicate these key presses to the automation controller. This can be communicated using TRAP: CONTROL CHANGE messages indicating which button has been pressed to adjust the selected light either up or down in brightness as illustrated in the bottom of FIG. 4b. The automation controller can then, in turn, make the requested changes to the overhead lighting component. The display can also include an indicator showing the current status of the overhead lighting which is updated as the lighting level is changed, for example, using additional lines on the display (not shown).

While the menu items shown in the above example use simple text strings, it should be appreciated that menus can take on a variety of formats. For example, menus can include title elements, graphical elements, multi-level hierarchy, non-hierarchical links, and other known elements.

Discussion of one exemplary detailed implementation of the automation controller will help to illustrate additional benefits of the present invention. A block diagram of an automation controller is shown in block diagram form in FIG. 5. The automation controller, shown generally at 60, can include a processor 62 which controls overall operation of the automation controller. Various processors are known which can be used in embodiments of the present invention, including for example, microprocessors and microcontrollers. Alternately, the processor may be implemented in logic, for example, using a field programmable gate array (FPGA) or application specific integrated circuit (ASIC). The home automation system may include one or more remote controls in communication with the automation controller. The automation controller includes means for communicating with the remote controller. For example, this means for communicating may be a wireless transceiver 64. Various wireless communications techniques are known which can be used in wireless transceivers, including for example, radio frequency, ultrasonic, optical, infrared, and ultraviolet transmission. Wireless transceivers for various communications techniques are known which can be used in embodiments of the present invention.

A control interface 66 can be included for communication with the automation components. As another option, communication with the automation components can also be via

US 10,877,623 B2

7

the wireless transceiver 64. The automation controller may also include a memory 68, such as random access memory (RAM), non-volatile RAM, read only memory (ROM), programmable ROM (PROM), erasable programmable ROM (EPROM), flash memory, and combinations of these types of memory and the like. The memory may be used to store the menu database 69, software to implement functions of the automation controller, remote control configuration parameters, and information for control of the automation components. The memory may be provided separately from the processor or included within the processor component.

By maintaining the menu database 69 at the automation controller 60, maintenance and upgrade of the home automation system can be enhanced. For example, as home automation network components are added or removed, menus can be modified to reflect the added or deleted capabilities. The menu database may be edited through a user interface (not shown) provided by the automation controller. As an alternative, the menu database may be updated remotely, such as through an internet connection (not shown) or connection (not shown) to a personal computer. Once the menu database is updated, these updates will be immediately available to remote controls, since the remote controls dynamically download menu items. This helps to avoid needing to separately reprogram the remote control(s) every time a menu update occurs, since updated menu items are automatically downloaded to the remote control when needed for display.

The automation controller 60 also includes means for extracting menu items from the database in response to request from the remote control. The extracted menu items can be in response to requests from the remote control, and the number of menu items and length of menu items based on the remote control capacity limits as described above. Various protocols for downloading menu item extracts can be used between the automation controller and the remote control, for example as described above.

Finally, a detailed example of a remote control is illustrated in FIG. 6, in accordance with an embodiment of the present invention. The remote control, shown generally at 70, includes a display 72 and user input controls 74. The display can include a touch screen which also serves as the user input controls. With a touch screen, various types of virtual or graphical controls can be implemented on the screen, such as a touch button, touch slider, touch rotary controls, and the like. For example, a representation of a slider control may be displayed using simple text, such as “<--0-->” where the ‘0’ corresponds to the current position of the control. The control can respond to touch movements on the screen, moving the ‘0’ back and forth to simulate the operation of a physical slider control.

A processor 76 is coupled to the display 72 and the user input controls 74, and can provide output to the display (such as displayed menus) and accept input from the user input controls (such as key presses, control activations, and the like). Various processor types can be used, including for example a microprocessor, microcontroller, discrete logic, FPGA, and ASIC.

The remote control 70 has a limited capacity memory 78, which can include RAM, NVRAM, ROM, PROM, EPROM, flash memory, and combinations thereof. The memory may be used to store limited numbers of menu items of limited length received from the automation controller, software to implement functions of the remote control, remote control configuration parameters, and information for operation of

8

the remote control. The memory may be provided separately from the processor or may be included within the processor component.

The remote control 70 includes means for communicating capacity limits to the automation controller. For example, the processor 76 can generate and transmit a CONFIGURATION PARAMETERS message to the automation controller using a wireless transceiver 80. Various wireless communications approaches are described above which can be used in the remote control.

The remote control 70 also includes means for dynamically downloading a limited number of menu items of a limited length from the automation controller. For example, downloading can be performed using the wireless transceiver 80 using protocols as described above in FIGS. 4a and 4b. Downloading can be performed as the user navigates through the menu items, for example as described above.

It should be appreciated that one advantage of the remote control 70 provided is that the amount of memory required can be quite small, as the entire menu database need not be stored on the remote control. Furthermore, since menus are downloaded to the remote control only when required by the user, the amount of bandwidth required over the wireless link to the automation controller can be relatively small. Accordingly, complex wireless techniques are not required and the wireless transceiver 80 can be relatively simple. Both of these features help to reduce the cost of the remote control.

In summary, embodiments of the present invention can help to simplify cost and operation of a home automation network. By keeping a database of menu definitions at the automation controller, database maintenance is centralized and streamlined. Manual updating of remote controls is avoided, since database extracts are dynamically downloaded to the remote control when needed. Relatively low bandwidth wireless communication between the remote control and the automation controller can be used, since menu downloads need not occur except when a user is navigating the menus. Remote control cost can be reduced, since less memory is used. Multiple alternate menu representations can be included in the database, with selection being based on the capability of the remote control requesting menu information. This can allow for adaptation of displayed menus and control selections to the capabilities of individual remote controls.

While the foregoing examples are illustrative of the principles of the present invention in one or more particular applications, it will be apparent to those of ordinary skill in the art that numerous modifications in form, usage and details of implementation can be made without the exercise of inventive faculty, and without departing from the principles and concepts of the invention. Accordingly, it is not intended that the invention be limited, except as by the claims set forth below.

The invention claimed is:

1. A method of operating a remote control having a display in a home automation network having an automation controller, comprising the steps of:

communicating remote control configuration parameters related to menu display capabilities of the remote control from the remote control to the automation controller, the remote control configuration parameters including a remote control memory capacity and a list of available physical controls present on the remote control, wherein the automation controller controls lighting dimmers and switches, audio/visual equipment and climate control in response to control messages

from the remote control, and wherein the automation controller is a network device separate from the remote control; and dynamically receiving database extracts from a menu database accessible by the automation controller at the remote control in response to menu navigation by a user, the database extracts defining displayable user interface menu elements and input control function mappings consistent with the remote control configuration parameters and comprising portions of the menu not currently displayed, wherein the automation controller sends the database extracts to the remote control in response to a request from the remote control for additional menu items that are not currently displayed but are adjacent to a displayed menu item, wherein the remote control sends the request for the additional menu items to the automation controller in response to a scroll movement to the additional menu items and based on an amount of cache memory available in the remote control, wherein a number of menu items and a length of each menu item are selected by the automation controller based on the remote control memory capacity, wherein the database extracts received are limited portions of the menu database, and wherein the input control function mappings are determined by the automation controller based on the communicated list of available physical controls present on the remote control.

2. The method of claim 1, further comprising the steps of: displaying a selection-type displayable user interface menu item on the remote control display when an input control is activated which has been function mapped to a menu navigation function; and displaying a control-type displayable user interface menu item on the remote control display when an input control is activated which has been function mapped to a menu selection function.

3. The method of claim 2, wherein the selection-type displayable menu item comprises text strings.

4. The method of claim 2, wherein the control-type displayable menu item comprises a control type chosen from the group consisting of a touch slider, a touch switch, a push button, a slider, a toggle switch, a rotary knob.

5. The method of claim 1, wherein the step of dynamically receiving database extracts comprises fetching a displayable menu item for immediate display.

6. The method of claim 1, wherein the step of dynamically receiving database extracts comprises fetching a displayable menu item for storage.

7. The method of claim 1, wherein the step of dynamically receiving database extracts comprises fetching a control type for immediate display.

8. The method of claim 1, wherein the remote control configuration parameters comprise a display width limit.

9. The method of claim 1, wherein the remote control configuration parameters comprise a display height limit.

10. The method of claim 1, wherein dynamically receiving database extracts comprises dynamically receiving database extracts in response to menu scrolling by the user, the method further comprising overwriting existing database extracts with the database extracts received in response to menu scrolling by the user.

11. The method of claim 1, wherein the input control function mappings include a mapping of push buttons being mapped to volume up and volume down functions.

12. The method of claim 1, wherein the input control function mappings include a mapping of a blank control being mapped to lights.

13. The method of claim 1, wherein the remote control requests the additional menu items from the automation controller using a GET ITEM message.

14. The method of claim 1, wherein in response to the scroll movement to the portions of the menu not currently displayed, the remote control makes a number of requests for the additional menu items based on the amount of cache memory available in the remote control.

15. A method of control in a home automation network having an automation controller and a remote control having a limited capacity memory and a limited capacity display, the method comprising the steps of:

- communicating remote control memory capacity parameters and a list of available physical controls present on the remote control from the remote control to the automation controller, wherein the automation controller controls lighting dimmers and switches, audio/visual equipment and climate control in response to control messages from the remote control, and wherein the automation controller is a network device separate from the remote control;
- downloading menu items and input control function mappings from the automation controller to the remote control, a number of menu items and a length of each menu item selected by the automation controller based on the remote control memory capacity parameters and the input control function mappings determined by the automation controller based on the communicated list of available physical controls present on the remote control; and
- displaying at least one of the menu items on a display on the remote control;
- changing the menu item displayed in response to user navigation commands; and
- downloading additional menu items from a menu database accessible by the automation controller in response to user navigation commands, wherein the additional menu items downloaded are limited portions of the menu database, wherein the automation controller sends the additional menu items to the remote control in response to a request from the remote control for the additional menu items that are not currently displayed but are adjacent to a displayed menu item, and wherein the remote control sends the request for the additional menu items to the automation controller in response to a scroll movement to the additional menu items and based on an amount of cache memory available in the remote control.

16. The method of claim 15, wherein the step of downloading additional menu items comprises:

- obtaining a next menu item not currently displayed on the remote control; and
- storing the next menu item in the limited capacity memory.

17. The method of claim 16, wherein the step of storing the next menu item comprises overwriting a menu item currently stored in the limited capacity memory.

18. A home automation network comprising:

- an automation controller coupled to a plurality of home automation network components for controlling the plurality of home automation network components, wherein the automation controller controls lighting dimmers and switches, audio/visual equipment and climate control in response to control messages from a

US 10,877,623 B2

11

remote control, and wherein the automation controller is a network device separate from the remote control; a menu database accessible to the automation controller, the menu database defining user-navigable control menus to allow user control of the home automation network components, the menu database including a plurality of the user-navigable control menus; and the remote control in communication with the automation controller, the remote control being arranged to dynamically download input control function mappings and limited portions of one of the plurality of user-navigable control menus in the menu database from the automation controller in response to a user navigating through the menu database using the remote control, wherein the automation controller sends the plurality of user-navigable control menus to the remote control in response to a request from the remote control for additional menu items that are not currently displayed but are adjacent to a displayed menu item, wherein the remote control sends the request for the additional menu items to the automation controller in response to a scroll movement to the additional menu items and based on an amount of cache memory available in the remote control, wherein a selection of the one of the plurality of user-navigable control menus is based on remote control memory capacity limitations communicated from the remote control to the automation controller, wherein a number of menu items and a length of each menu item are selected by the automation controller based on the remote control memory capacity limitations, and wherein the input control function mappings are determined by the automation controller based on a list of available physical controls present on the remote control communicated from the remote control to the automation controller.

19. The home automation network of claim 18, wherein the remote control further comprises:

- a display for displaying a selection from the menu database; and
- a user-input control for accepting user input.

20. The home automation network of claim 18, wherein the menu database comprises a plurality of text strings.

21. The home automation network of claim 18, wherein the menu database comprises a plurality of control type identifiers.

22. The home automation network of claim 18, further comprising the plurality of home automation network components.

23. An automation controller for a home automation network for use with a remote control, the automation controller comprising:

- a database storage having plural navigable menu items stored therein;
- a processor for extracting menu items and input control function mappings from the database in response to a request from the remote control for additional menu items that are not currently displayed but are adjacent to a displayed menu item, wherein the remote control sends the request for the additional menu items to the

12

automation controller in response to a scroll movement to the additional menu items and based on an amount of cache memory available in the remote control, wherein a number of menu items and length of menu items is based on remote control memory capacity limits, wherein the extracted menu items are limited portions of the database, and wherein the input control function mappings are determined by the automation controller based on a list of available physical controls present on the remote control; and

- a transceiver for communicating extracted menu items to the remote control;

wherein the automation controller controls lighting dimmers and switches, audio/visual equipment and climate control in response to control messages from the remote control, and wherein the automation controller is a network device separate from the remote control.

24. A remote control for a home automation network for use with an automation controller having control of the home automation network, the remote control comprising:

- a display device for displaying menu items to a user;
- a transceiver for communicating a remote control memory capacity and a list of available physical controls present on the remote control to the automation controller and for dynamically downloading input control function mappings and a limited number of menu items of a limited length from the automation controller as a user navigates through the menu items, wherein the automation controller sends the input control function mappings and the limited number of menu items to the remote control in response to a request from the remote control for additional menu items that are not currently displayed but are adjacent to a displayed menu item, wherein the remote control sends the request for the additional menu items to the automation controller in response to a scroll movement to the additional menu items and based on an amount of cache memory available in the remote control, wherein the automation controller controls lighting dimmers and switches, audio/visual equipment and climate control in response to control messages from the remote control, wherein the automation controller is a network device separate from the remote control wherein the number of menu items and the length of each menu item are selected by the automation controller based on the remote control memory capacity, wherein the limited number of menu items are limited portions of a menu database accessible by the automation controller, and wherein the input control function mappings are determined by the automation controller based on the list of available physical controls present on the remote control.

25. The method of claim 13, wherein the GET ITEM message comprises a menu list ID and a menu item index for the additional menu items.

26. The method of claim 13, wherein the automation controller responds to the GET ITEM message with a SET ITEM message comprising the additional menu items.

* * * * *

Exhibit B



US008375137B2

(12) **United States Patent**
Bowman

(10) **Patent No.:** **US 8,375,137 B2**
(45) **Date of Patent:** **Feb. 12, 2013**

(54) **SYSTEM AND METHOD FOR STREAMING AUDIO USING A SEND QUEUE**

7,532,805 B2 * 5/2009 Okada et al. 709/231
7,970,875 B1 * 6/2011 Shaffer et al. 709/223
2005/0111662 A1 * 5/2005 Fiedler 380/201

(75) Inventor: **Timothy D. Bowman**, Salt Lake City, UT (US)

FOREIGN PATENT DOCUMENTS

(73) Assignee: **Control4 Corporation**, Salt Lake City, UT (US)

JP 2004096684 A * 3/2004
KR 739320 B1 * 7/2007
WO WO 9854646 A2 * 12/1998

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 185 days.

OTHER PUBLICATIONS

(21) Appl. No.: **12/507,754**

Lertsutthiwong et al. "Scalable Video Streaming for Single-Hop Wireless Networks Using a Contention-Based Access MAC Protocol." *Advances in Multimedia*. Hindawi Publishing Corp. Jan. 2008. vol. 2008, Issue 2, 21 pages.*
NN8607787. "Interface for Playing Back Recorded Audio Data." IBM Technical Disclosure Bulletin. Jul. 1, 1986. vol. 29, Issue 2. 4 pages.*
NB9002113. "Dynamic Reformatting and Merging of Data Streams." IBM Technical Disclosure Bulletin. Feb. 1, 1990. vol. 32, Issue 9B. 8 pages.*

(22) Filed: **Jul. 22, 2009**

(65) **Prior Publication Data**

US 2010/0023639 A1 Jan. 28, 2010

* cited by examiner

Related U.S. Application Data

(60) Provisional application No. 61/082,807, filed on Jul. 22, 2008.

Primary Examiner — Jeffrey R Swearingen

(51) **Int. Cl.**
G06F 15/16 (2006.01)

(74) *Attorney, Agent, or Firm* — Thorpe North & Western LLP

(52) **U.S. Cl.** 709/231; 709/219

(58) **Field of Classification Search** None
See application file for complete search history.

(57) **ABSTRACT**

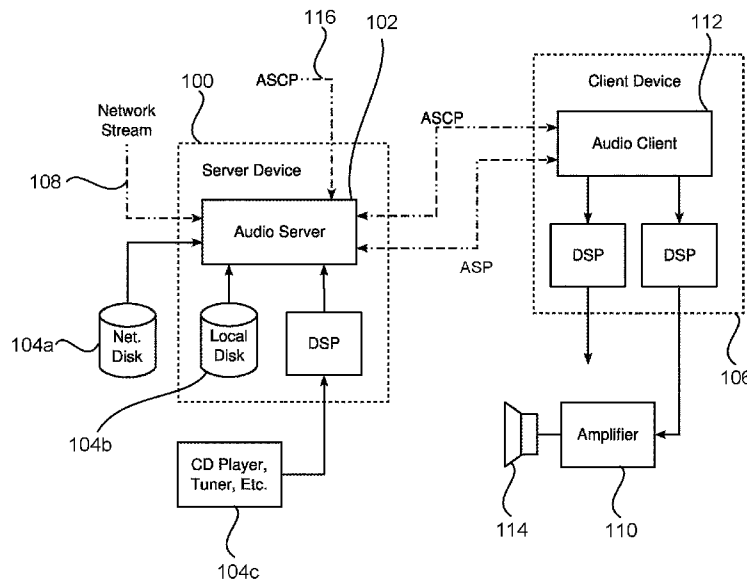
A system and method are provided for preparing a streaming media system for initial presentation of a media stream. The system includes a media server configured to send out media packets for a media stream at periodic clocked intervals. A framer can be located with the media server to divide the media stream into media packets. A media client is also provided to receive the media packets for the media stream from the media server. A send queue can be located in the server. The send queue can be configured to store a defined length of programming from the media stream, and the send queue can immediately fill a client's user buffer when an activation event occurs.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,586,264 A * 12/1996 Belknap et al. 709/231
5,640,597 A * 6/1997 Noguchi et al. 709/231
6,216,173 B1 * 4/2001 Jones et al. 709/231
7,058,721 B1 * 6/2006 Ellison et al. 709/231
7,142,509 B1 * 11/2006 Rovner et al. 370/230

20 Claims, 11 Drawing Sheets



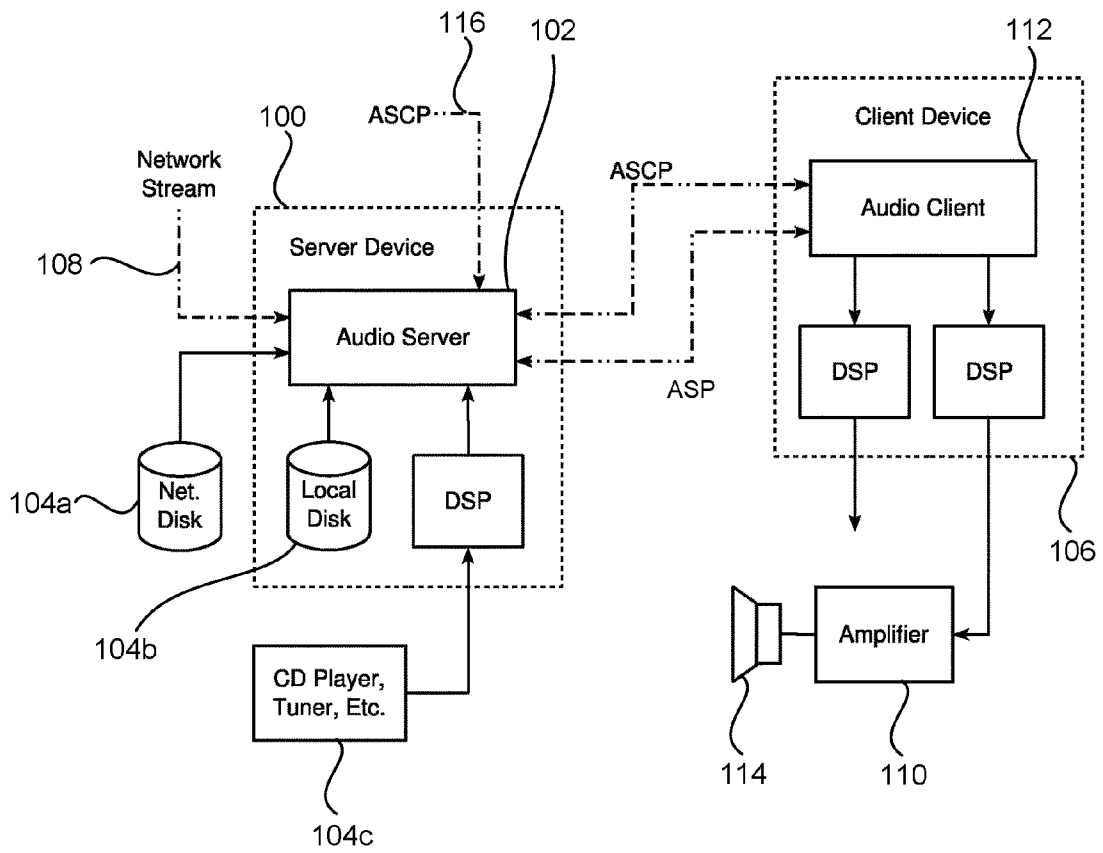


FIG. 1

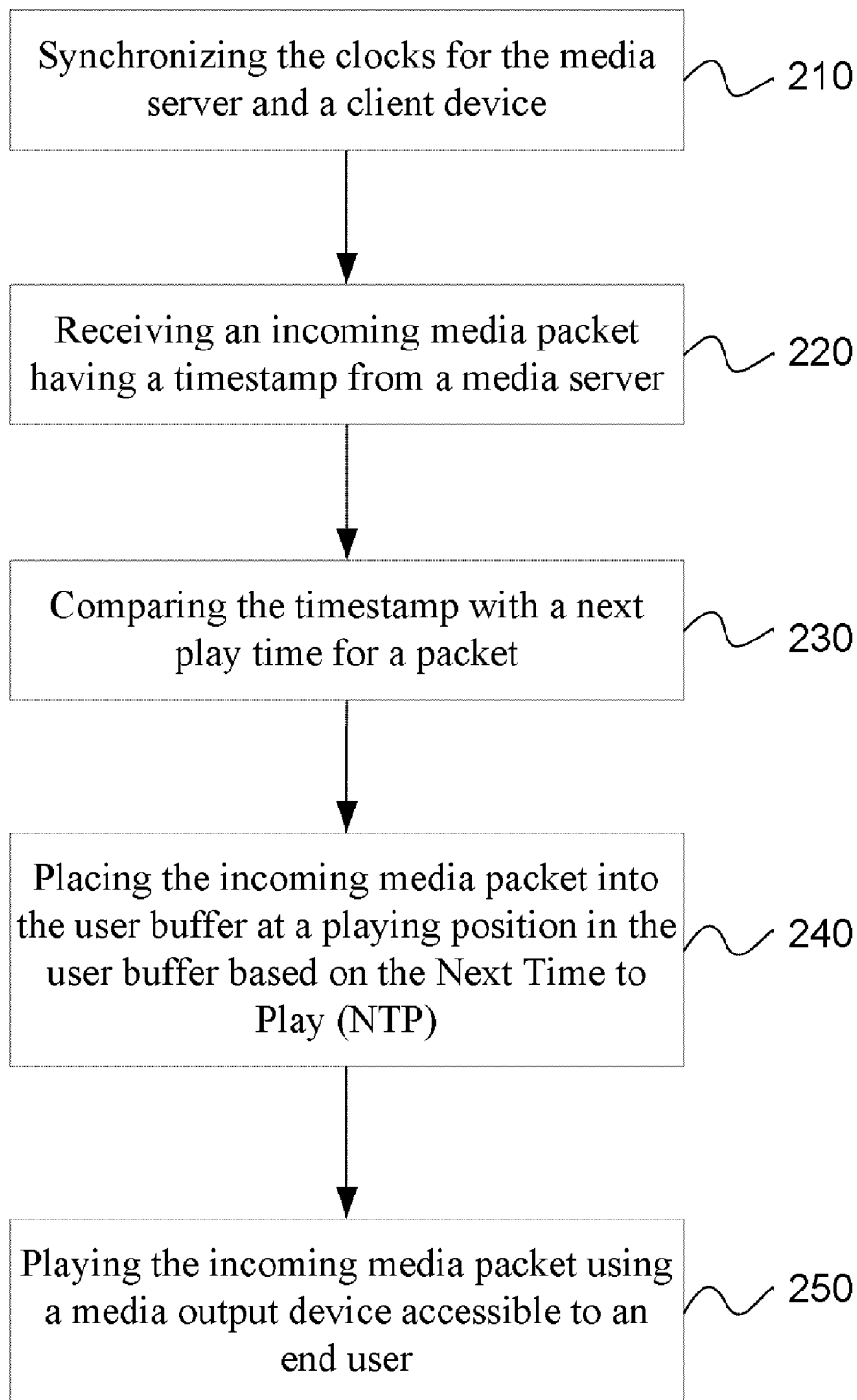


FIG. 2

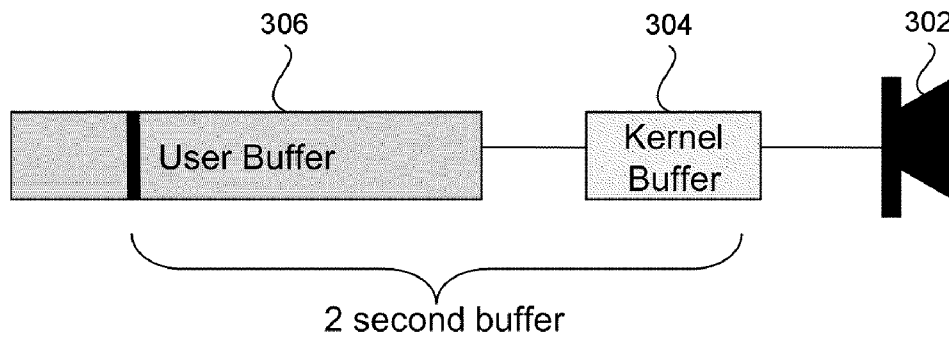


FIG. 3a

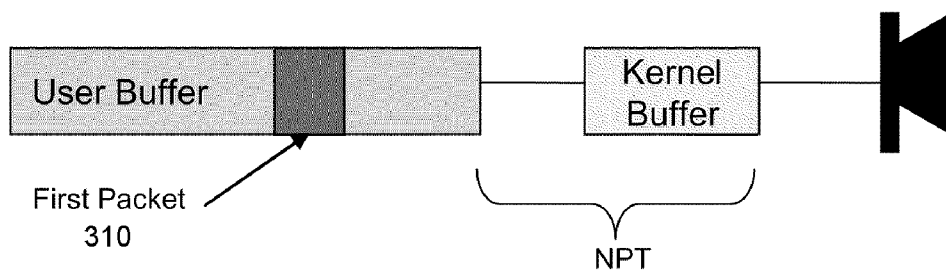


FIG. 3b

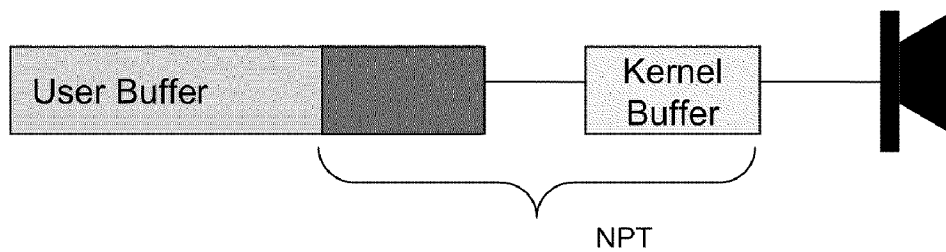


FIG. 3c

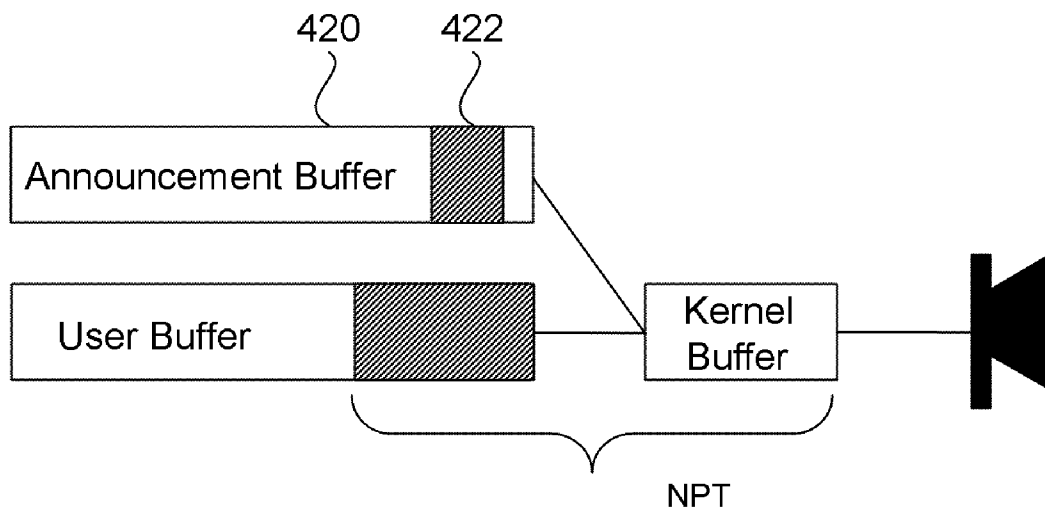


FIG. 4

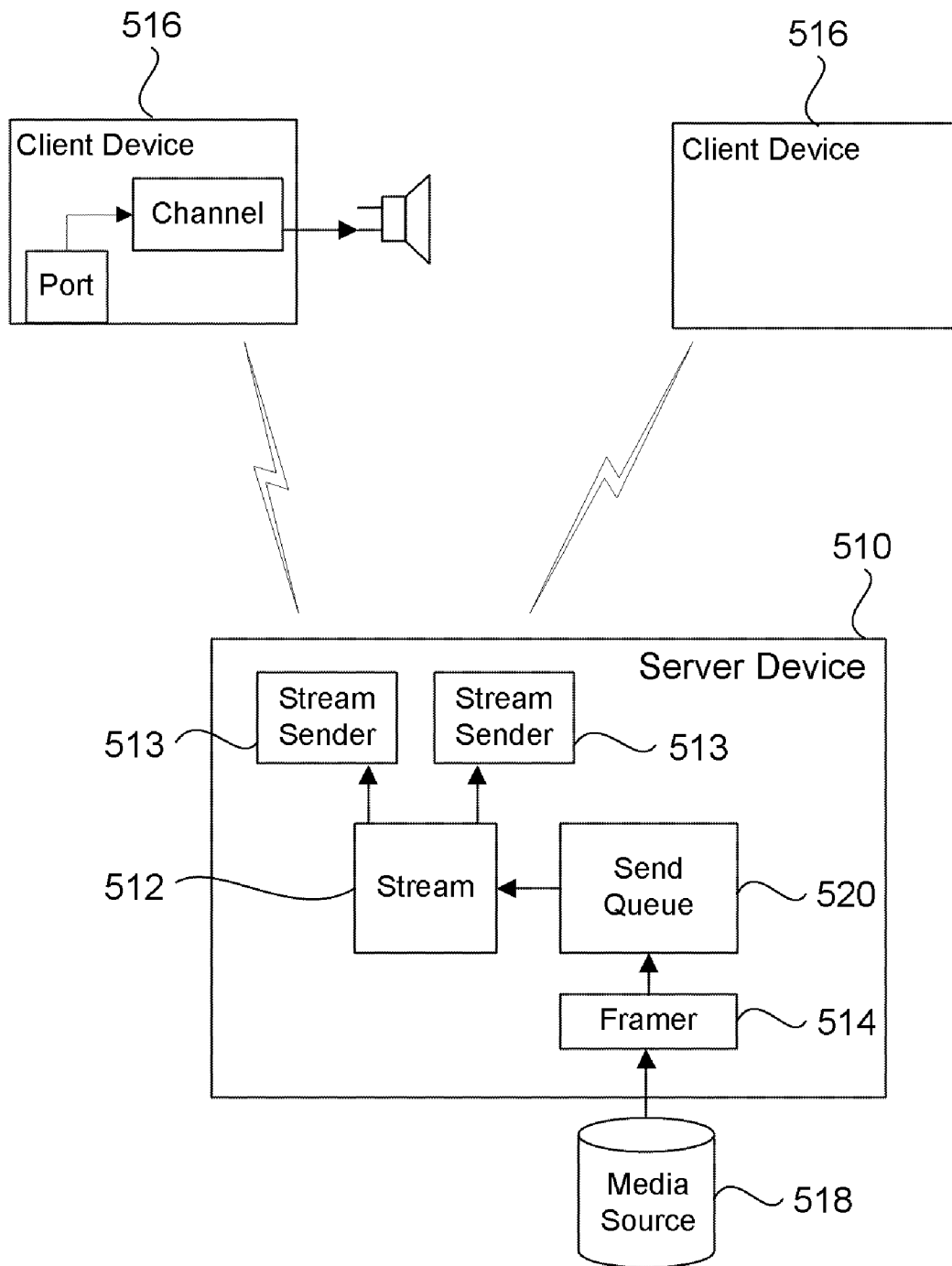


FIG. 5a

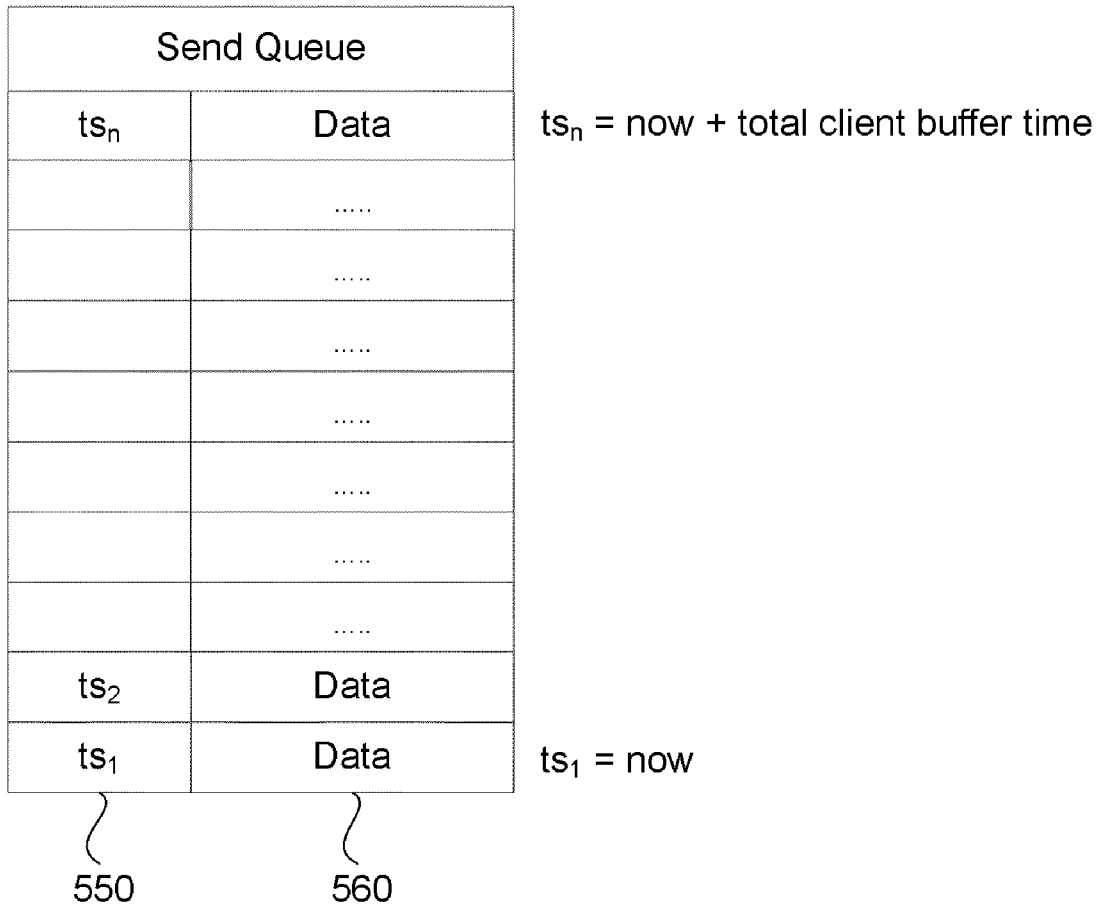


FIG. 5b

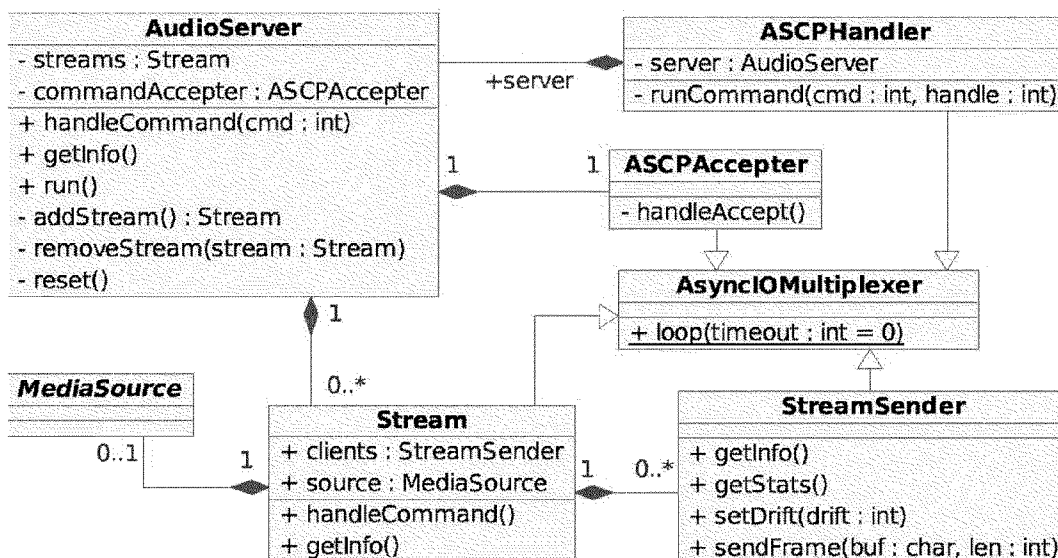


FIG. 6

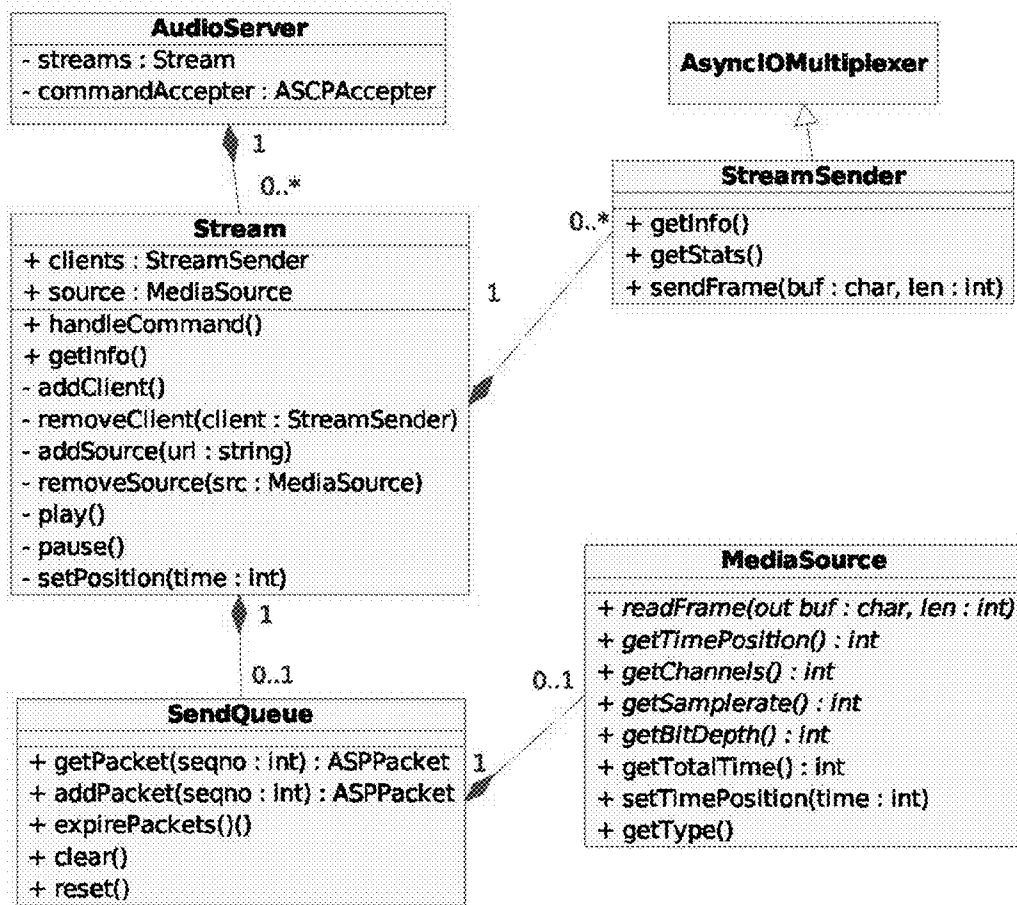


FIG. 7

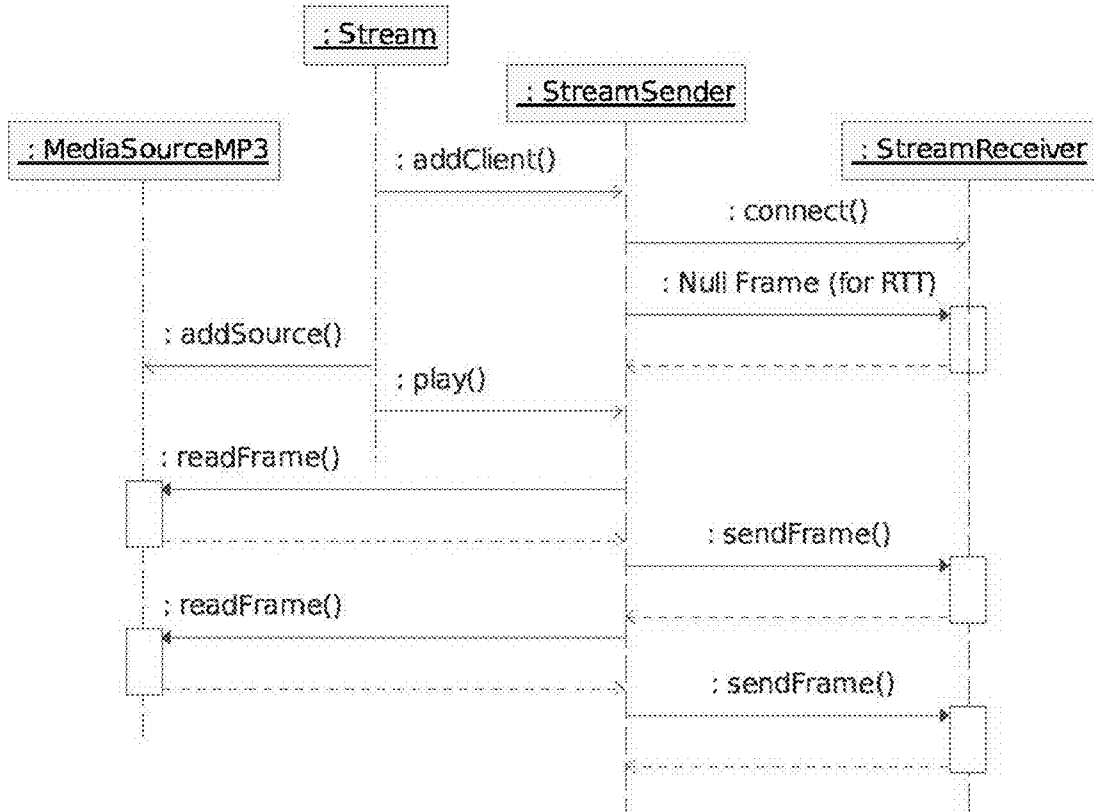


FIG. 8

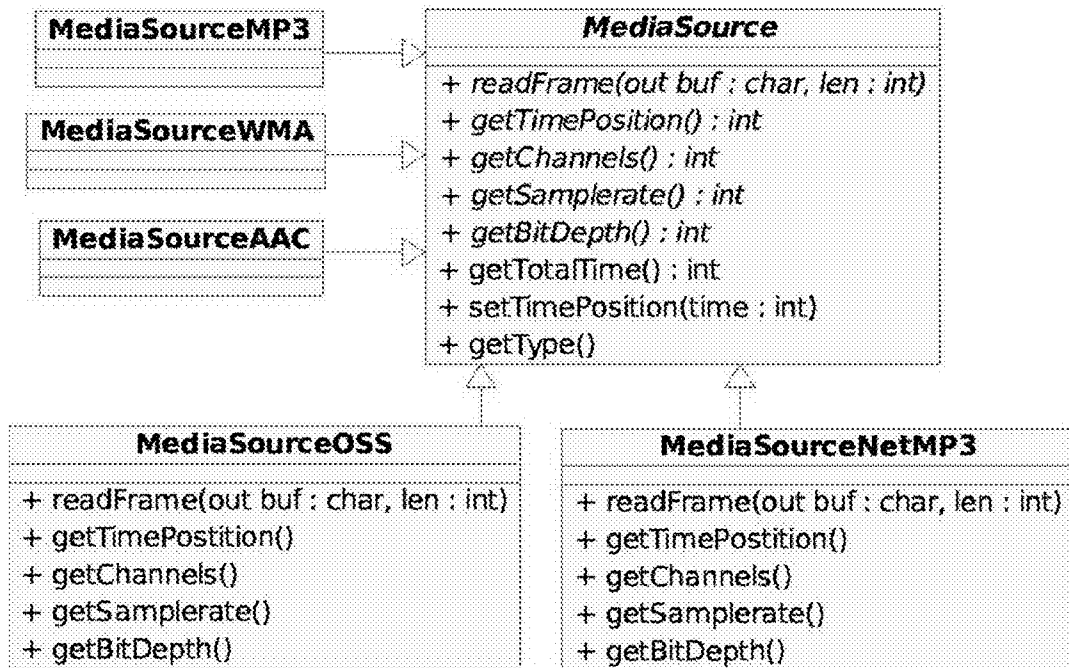


FIG. 9

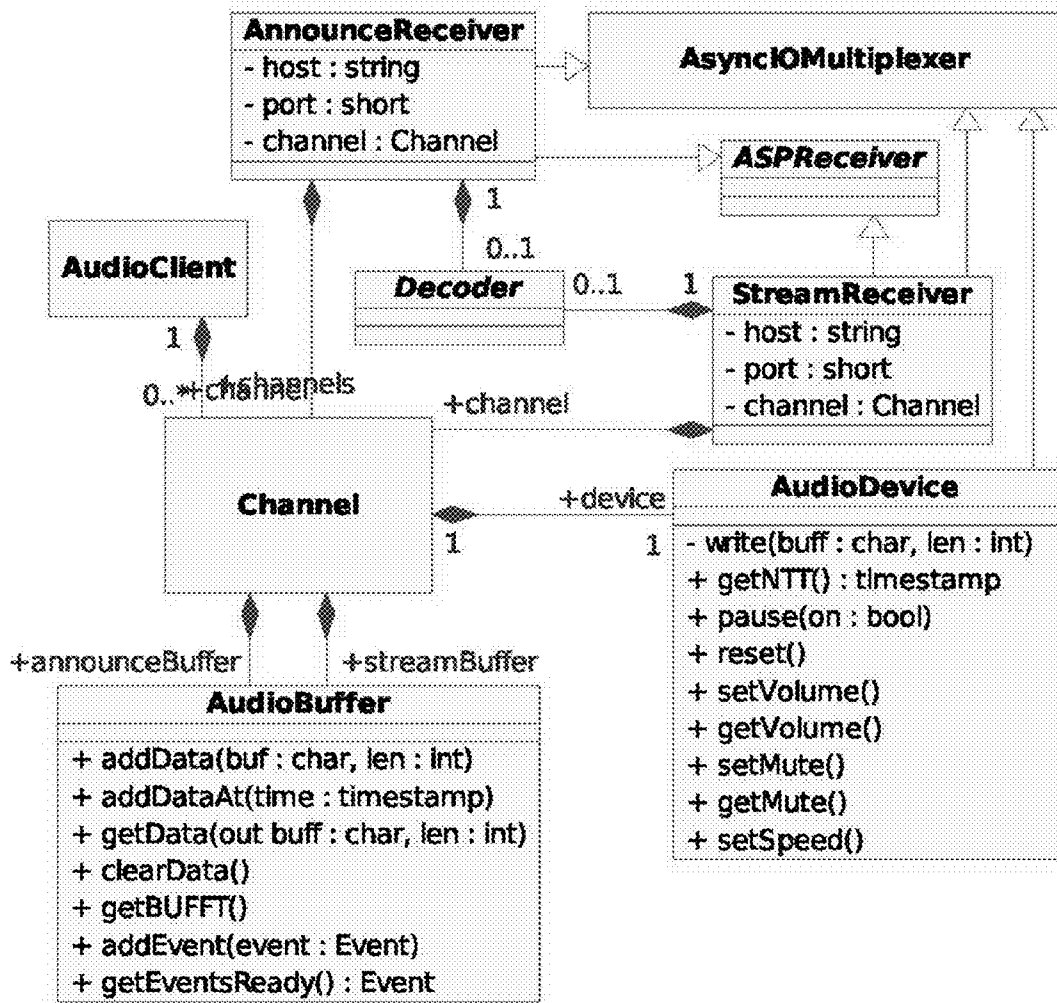


FIG. 10

US 8,375,137 B2

1

SYSTEM AND METHOD FOR STREAMING AUDIO USING A SEND QUEUE

PRIORITY CLAIM

Priority is claimed to U.S. Provisional Patent Application Ser. No. 61/082,807, filed Jul. 22, 2008, which is hereby incorporated herein by reference in its entirety.

BACKGROUND

Wired and wireless networks are becoming a significant method for distributing media files and other information to end users. It is now possible to distribute music, video, games, and other multi-media information to computers, thin-clients, cell phones, and virtually any network capable device within a wireless or wired network.

Streaming of data files or "streaming media" refers to technology that delivers sequential content at a rate sufficient to present the media to a user at the originally anticipated playback speed without significant interruption. Streamed data may be stored in device memory until the data is played back and then the streamed data can be subsequently deleted. In most streaming systems, the user is able to start accessing the requested content without waiting for the entire media file to be completely transferred over the host network.

However, the audio/video quality that can be received for a streaming presentation can be constrained by the available bandwidth and latency of the network connection. The constrained bandwidth and latency may cause congestion of the streaming content, slowing down the reception of the requested content. Also, a network failure may prevent the reception of the streaming content. These deficiencies may impact the effectiveness of streaming as it delivers content on demand from previously recorded broadcasts or content from live broadcasts.

In particular, the streaming deficiencies may impact home automation networks. A home automation network connects electronic devices that control lighting, monitor appliances, provide high quality entertainment, Internet connections, and supply a multitude of functions for a home. In these home automation networks, media files can be streamed from a central controller to client devices or streams can be sent between devices. In situations where a stream is sent to multiple clients within a single building, the system can substantially synchronize the output of the media streams to avoid any noticeable delays that may be heard when the audio or video files are played simultaneously across multiple rooms.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram illustrating selected components of a server device and client device in accordance with an embodiment;

FIG. 2 is a flowchart of a method of synchronizing audio streaming in an embodiment;

FIG. 3a is a block diagram illustrating a user buffer and a kernel buffer in an embodiment of a client device;

FIG. 3b is a block diagram illustrating a packet being inserted into a user buffer in an embodiment of a client device;

FIG. 3c is a block diagram illustrating a user buffer and a kernel buffer with a queue of audio packets in an embodiment of a client device;

FIG. 4 is a block diagram illustrating an announcement buffer that is used with a user buffer in an embodiment of a client device;

2

FIG. 5a is a simplified block diagram illustrating the use of a send queue in a server system in an embodiment;

FIG. 5b illustrates an example configuration of a send queue;

FIG. 6 illustrates a software module that can be used in an embodiment of a server system;

FIG. 7 illustrates software modules that can be used for a signal media stream source for one or more clients in an embodiment;

FIG. 8 illustrates software modules configured to open, close, and read different types of media in an embodiment;

FIG. 9 is a block diagram illustrating media source software modules in an embodiment; and

FIG. 10 illustrates software modules configured to manage connections made to the client stream or announcement ports in an embodiment.

DETAILED DESCRIPTION

Reference will now be made to the exemplary embodiments illustrated in the drawings, and specific language will be used herein to describe the same. It will nevertheless be understood that no limitation of the scope of the invention is thereby intended. Alterations and further modifications of the inventive features illustrated herein, and additional applications of the principles of the inventions as illustrated herein, which would occur to one skilled in the relevant art and having possession of this disclosure, are to be considered within the scope of the invention.

A system and method for audio streaming are provided to distribute audio media sources or streams to one or more audio clients, to ensure substantial synchronization of audio outputs when multiple clients are playing the same stream, and to provide basic stream transport and audio control commands. While this system will be described primarily as an audio system, other types of streaming media such as video, streaming computer graphics or other media streams can also be sent through the streaming system.

Two components of the system may be the audio server and audio client. FIG. 1 illustrates that the audio server 102 reads various kinds of audio sources 104a-c, divides the data into frames suitable for streaming over a network, and then sends the data stream or audio stream to one or more audio client devices 106. Then the audio stream can be played back using a signal output device such as an amplifier 110 and a speaker 114. The audio server can broadcast packets through a wireless or wired network, and software buffering can also be provided by the system. Several general types of media sources can be streamed by the audio server.

One type of media source may include an MP3, WMA, M4A, or another type of electronically encoded media file accessible as part of the file system of the computing platform 100 running the audio server software. Examples of sources are MP3 files stored on the local hard disk drive 104b of a media controller, M4A files on a USB stick drive inserted in a home theater controller, or songs on an iPod® mounted as a USB device on a speaker device or client device.

Another type of media source can be the audio inputs of any electronic device or computing product that provides media streams 104c, such as a Linux OSS (/dev/dspN) device. This capability makes it possible to distribute and synchronize the analog, digital, or optical outputs of devices like CD players and tuners when they are connected to a media controller, home theater controller, or audio output device. Such inputs may also include a microphone on a mini touch panel, which can be used to implement an intercom system. Another

US 8,375,137 B2

3

type of media source is a remote network stream **108**, such as internet radio, which can be read by the audio server **102** and distributed to clients.

Audio frames can be sent to clients using Audio Stream Protocol (ASP), described in more detail later. Real-time Transport Protocol (i.e., RFC 3550) can be used to allow inter-operability between third party devices or software and audio servers and clients. Because of issues with User Datagram Protocol (UDP) on Wi-Fi networks, Transmission Control Protocol (TCP) is more effective for the network transport but UDP or other protocols may be used if the network conditions are appropriate. Real-Time Transport Protocol (RTP) is less suited for a streaming protocol such as TCP, and is preferably encapsulated in a wrapper protocol. RTP is also less suited to a real-time synchronization method and may need to use the Synchronized Source (SSRC) field to provide a high-resolution timestamp for clients, instead of the random source identifier used by the RTP specification. For this reason, and also to facilitate the streaming of multiple audio frames as a group, the ASP protocol may be used.

An audio client **112**, as in FIG. 1, can be any computing device running audio client software and a device that has analog outputs, digital outputs, optical outputs, or media presentation outputs, and network connectivity. The audio client can receive ASP or RTP packets, decode the data if necessary, and send the data to an output device such as an amplifier **110** and speaker **114**. The timing of the audio output may be adjusted by the client according to information sent by the server in the ASP or RTP header extension. This timing information can be calculated by the server based on the presentation rate of the media and the server's system clock.

The audio server and client do not generally handle high level tasks such as locating and scanning media, building playlists, or selecting a group of clients. These functions can be controlled by the audio driver on the system running a director software program and by using the Audio Server Control Protocol (ASCP) (discussed more later) **116**. Both the server and client listen for connections on a pre-determined TCP port, and both the server and client can accept multiple connections. Server commands are available to set up client endpoints, load media, and operate the transport controls. Client commands are available for client-specific settings like volume or tone and for retrieving client status. The audio system may also be controlled by manually entering commands in a telnet terminal session.

The streaming system can also provide time synchronization. Every computing device running an audio client and/or server has a system clock that is synchronized with every other audio device on the local network using the Network Time Protocol (NTP) (i.e., RFC 1305).

FIG. 2 illustrates a method of synchronizing audio streaming and FIGS. 3a-3c illustrate the client buffers that may be used in an embodiment of the method. The method includes the operation of synchronizing clocks of the media server and a client device. The synchronization can take place using the Network Time Protocol (discussed in detail later) or another clock synchronization method used in networks, as in block **210**.

Another operation is receiving an incoming media packet having a timestamp from a media server, as in block **220**. The media packet can be received by the audio client from the audio server. One example of a network that may be used to send the packet between the server and client is an 802.11x wireless network but other types of wireless or wired networks (e.g. Ethernet at a sufficient speed) may also be used.

When the client receives the media packet, then the client can compare the timestamp of the packet being received with

4

the time that the next frame of data will actually be played out on the device or Next Play Time (NPT), as in block **230**. The incoming media packet can be placed into the user buffer at a playing position in the user buffer based on the (NPT), as in block **240**. Incoming media packets can then be played using a media output device **302** (FIG. 3) accessible to an end user, as in block **250** (FIG. 2). For example, the end user will then be able to hear the audio using a speaker and/or view any video that might be displayed on a video screen.

Referring to FIG. 3a, the NPT (Next Play Time) is defined by the number of packets to be played in the kernel buffer **304**, the number of packets in the user buffer **306**, and the current system time. If the timestamp for the incoming packet is located within the expected window for the NPT, then the packet is added into the play queue at the appropriate position as illustrated in FIG. 3b where the first packet **310** in a stream is shown being inserted into the appropriate location based on its timestamp. The first packet in the stream is placed into the user buffer with a defined amount of quiet time before it begins to play. This allows the stream to start playing without any stuttering or jitter.

FIG. 3c shows the user buffer after a number of packets have been added into the user buffer. Then an additional incoming packet can be added to the end of the queue if the timestamp is within the appropriate time window as compared to the NPT at the end of the queue. In short, the incoming media packet is placed into the user buffer at a time position in the user buffer based on the timestamp of the incoming media packet as compared to the client clock and the NPT, keeping in mind that the client and server clocks are synchronized, and the server clock is used to set the packet timestamp.

Since each of the clients have different playout clock crystals, each of the clients can calculate a drift co-efficient based on the expansion or contraction of their user buffer **306** as more packets are added and played. This drift co-efficient can be used to independently adjust the client's timing by upsampling or downsampling. In addition, each client may also re-sample the incoming data to convert the data sample rate to match the sampling rate used by the user queue and the kernel queue.

Since the clocks of all the devices are calibrated, then the clients may play the media packets to within a predefined number of milliseconds of each other. For example, +/-20 milliseconds is within a tolerance that enables the media streams to sound like each client is playing the media stream at exactly the same time. However, other tolerance ranges may be used.

The system illustrated in FIGS. 3a-c, further illustrates that packets are frequently transferred from the user buffer to the kernel buffer to enable the packets to be played using an audio speaker for the client. A kernel buffer process can be configured to frequently move available packets from the user buffer to the kernel buffer. The kernel buffer process can check the user buffer at defined periodic intervals and/or whenever there are available processor clock cycles.

These separate buffers enable the system to be paused more quickly and easily. Once packets are put into the kernel buffer they cannot be revoked and will be played by the output hardware. However, if a user presses the pause button in one client or at the server, then a packet is sent via the server to flush the entire user buffer on each client. As a result, the kernel buffer will play silence until more data is sent to the user buffer.

The streaming media system can also include an announcement buffer as illustrated in FIG. 4. Announcements or brief audio segments mixed with existing audio streams are

US 8,375,137 B2

5

handled by the audio client on a separate TCP port. The announcements can be messages such as the doorbell, voice announcements, an intercom or synthesized error messages. When audio data is sent to this port, the announcement data is played on the specified audio output. If a stream is currently running on that output, its volume is “ducked” while the announcement plays, and synchronization of the main stream is not interrupted. The term “duck” is generally defined here to mean that the announcement is either blended into the main stream at some ratio or that the announcement stream replaces the main stream for a short duration. For example, the announcement stream may be mixed at 75% of its original volume and the main stream may be mixed at 25% of its original volume.

FIG. 4 illustrates that the announcement media packets 422 can be received into an announcement buffer 420 and be provided with some minimal amount of quiet space before the announcement. Announcements may or may not be synchronized like typical media streams, the announcements are put right into the queue without checking the timestamps. In the case of unsynchronized announcements, this means that the server does not throttle the announcements like it does the other media streams. Of course, an embodiment with synchronized announcements can be provided, if desired.

The kernel process can periodically check to see if packets are in the announcement buffer. If packets are in the announcement buffer, then the announcement media packets can be mixed with packets from the user buffer and placed into the kernel buffer to be played. A quiet period can be inserted before and/or after the announcement packets. Providing a quiet period surrounding announcements helps the end user to better understand the announcements.

FIG. 5a is a simplified block diagram illustrating the use of a send queue in a server device. The server device 510 is configured to prepare a media stream for presentation to an end user through a client device. A media stream 512 is configured to send out media packets to multiple clients at periodic clocked intervals through stream senders 513. The stream senders can send the media packets out using either a wireless or wired medium. The clocked intervals may be set to send out packets at the speed the media stream is expected to playback.

A framer 514 is located with the media server to divide the media stream into media packets. The framer can receive a media stream from a media source, such as an audio source 518. A media client device 516 can receive the media packets for the media stream from the stream sender 513 through a media port.

A send queue 520 can be provided in the server. The send queue may be configured to store a defined length of programming from the media stream. This means that at any point in time the send queue will contain packets that cover a time window for the media program. For example, 3 seconds of audio and/or video can be contained in the send queue. The audio server can then use the send queue to immediately fill a client’s user buffer when an activation event occurs.

Examples of activation events include a startup event or a reset event. When a startup event is received by the server from a client, then all the playable media stream information in the send queue can be sent to the client device immediately. In other words, the data from the send queue is sent to the client device without a clocking delay to allow the client device to startup quickly without waiting for a series of packets to be sent through the normal clocking cycles.

The send queue may also have housekeeping or garbage collection applied from time to time. This means that as the time to play passes and the packets are not usable, then the

6

packet can be erased or marked to be overwritten. As a result, the send queue can be implemented as a ring buffer or another configuration.

FIG. 5b illustrates an example configuration of a send queue. Each packet in the send queue buffer can have a timestamp 550 and then a data payload 560. The buffer may include a queue of packets that starts at a given time represented by ts_1 which is the timestamp for the current time or “now.” Then packets may be contained in the send queue that is contained within a few second window of “now”, and this is depicted as the packets ranging up to ts_1 or “now plus the total buffer time.”

The following discussion describes an example of time synchronization. Unlike most media streaming applications, which are concerned with synchronizing media from different sources, such as audio and video, and combining them into a synchronized presentation for a single client (or multiple clients in separate physical locations), the server may present a single media stream to a group of clients in the same physical area, which may be at various network distances from the server. If all clients in the group begin media output as soon as their stream buffer is ready, there may be audible or even visible artifacts in the sound or video when a listener or viewer receives the output from two or more clients, such as when the receiving party is positioned between two rooms.

Even if all of the audio clients are precisely synchronized, there is a delay associated with the distance between the listener and the sound source on the order of 0.9 ms/ft, due to the speed of sound in air. In a large home, this natural spatial delay could be as much as 30 or 40 ms between rooms, and it is desirable that the audio streaming subsystem does not introduce additional delays of this magnitude. The design goal for a time synchronization module is to keep clients synchronized to within a few multiples of the Linux system clock granularity (10 ms).

Network Time Protocol (NTP) can be used to synchronize the system clocks between all audio servers and clients. One system (the one running director software) can be setup as the master NTP server, and all other systems on the local network will be configured to synchronize to that server. The master system may also be synchronized to an external NTP server, but will fall back to its own system clock if network connectivity is interrupted or unavailable. If the time is set manually on the master system, clients will be notified to reset their clocks immediately, rather than wait (possibly many hours) for the NTP clients to re-sync their clocks. This will minimize the effect of time setting on audio streaming, although ample warning should be given to users that setting the system clock may result in temporary loss of audio synchronization.

One advantage of NTP is the ability to adjust clock skew parameters in the kernel, so the clocks on each system are not continuously drifting and being bumped forward or back by large increments. After a settling-in period, a drift correction factor is automatically calculated for each system, and written to a drift file that preserves the setting between re-boots.

The standard NTP implementation may have disadvantages that make an alternate synchronization technique attractive. First, the size of the executable program for synchronization may use over 2 MB of memory. This is a potential problem on small “embedded” devices such as an audio output device that does not have swapping capability. Second, NTP can take a significant amount of time to “stabilize” the system clock (on the order of tens of minutes or even several hours when network latency varies wildly, like in a poor Wi-Fi environment). This is especially true for devices just “out of the box”, whose local drift settings have not yet been tuned by the NTP daemon. Two other techniques may also be used for

time synchronization: 1) Reference Broadcast and 2) a subset of NTP tailored to the needs of the audio client and server.

The following discussion describes two network protocol modules and a utility class that can be shared between the audio server and audio client. Audio Stream Control Protocol (ASCP) is a simple protocol for handling requests and responses between audio clients and audio servers, and for controlling the audio streaming system from outside applications like a director software program. The director software is a program that is configured to control, route, and convert the many inputs that may enter a server. ASCP is an ASCII text protocol, and the commands, responses, and data are sent as lines of text terminated with <CR><LF>. This allows terminal access to the server and clients for testing and debugging.

Tokens in command and response lines are case-insensitive and are separated by white space. Commands must be prefixed with an alphanumeric tag, such as “A001”, generated by the client, and responses to that command are then prefixed with the same tag. If the tag or command is missing, the response is tagged with a “*”. Command parameters are separated by white space (and quoted if they contain spaces) and are case-sensitive.

Messages that are not associated with a command can be returned by the server, and are prefixed with the tag “SYS”, which indicates that the data is an “out of context” system message such as a low memory warning, or an informational message such as “EOM” (end of media). Command completion is indicated by responses with response codes as the first token after the tag, followed by the name of the command: OK, NO, or BAD.

One limitation of some previous audio server/client designs has been the reliance on raw RTP packets sent over UDP. UDP can make it easy for the client to read individual audio frames and pass them to the decoder. However, using a standard RTP header, which has no length field, can make the server send one packet at a time. Sending very small packets (one frame of audio data) at a high frequency may be problematic on 802.11 networks, where the throughput may be good, but the latency when sending periodically to a particular client may be very high. This problem can be aggravated by frequent 802.11 multicasts and Address Resolution Protocol (ARP) traffic, which causes frequent rate shifting in a WiFi network. The improved protocol allows a variable number of frames to be sent to clients at one time and can significantly improve streaming over 802.11 networks.

By using TCP for the data connection and using the Audio Stream Protocol (ASP) defined here, the server can be configured to send a variable numbers of frames, and the frames can be delimited by the protocol so it is not necessary for the client to re-frame the received data before passing it to the decoder. In one example embodiment, the ASP protocol header may have a 100 microsecond resolution timestamp, payload type, and sequence number, plus client control information and a length for the entire frame group. Each frame in the group is preceded by a two byte frame length, so the client can easily pull one frame at a time from the stream and pass it to the decoder.

The AsyncIOMultiplexer class allows for fine-grained control of input and output in each type of subclass (ASCPHandler, Server, and Client), but handles the details of selection and file descriptor sets, closing sockets on errors, etc. The AsyncIOMultiplexer eliminates the large, complex select loop that would otherwise be required to multiplex the audio server or client.

The ASCPHandler class is a subclass of AsyncIOMultiplexer, that is setup to receive and tokenize ASCP commands

from a client connection, map them to the internal command table, and call handleCommand() in the AudioServer and AudioClient main objects. All details of the external ASCP command protocol are handled in this class, so that the protocol may be changed without affecting the various client and server objects that implement commands. A number of classes used in the audio server and client are simple subclasses of AsyncIOMultiplexer that are set up as listening sockets and override the handleAccept() command. They are used to establish remote connections.

Audio Server

This section describes an example design of the audio server in more detail. The audio server is a daemon process, started automatically at system initialization, and runs on any home automation device or computing device that can be used as a media source.

Referring to the UML class diagram in FIG. 6, the main object in the audio server (first object to be instantiated) is the AudioServer. The AudioServer creates a single ASCPAccepter object derived from the AsyncIOMultiplexer class, and its run() method runs the loop() function in the AsyncIOMultiplexer class. The ASCPAccepter listens for incoming connections on a port (e.g., port 6000).

Stream objects represent a single Media source streaming to one or more audio clients. The AudioServer adds one or more Stream objects, and adds a StreamSender object to the Stream for every audio client that will receive the stream data. StreamSender objects are derived from AsyncIOMultiplexer, and are consequently added to a global socket map for I/O. AudioServer maintains a list of Stream instances, and passes non-AudioServer commands from the ASCPHandler to the correct Stream object.

Media is then associated with a Stream by calling a method in the Stream that creates a MediaSource object implementation based on the file or device type passed in the media URL. MediaSource objects provides a frame reading method, and may also provide position control methods, as well as information about the audio samples that are needed to build ASP headers.

On an incoming connection, an ASCPHandler object is created by CommandAcceptor, which handles incoming ASCP commands in a server control session. The ASCPHandler then maps the incoming command to a command table, and calls the handleCommand() function in AudioServer. If the command is a Stream command, it is passed to a Stream object identified in the command. If it is one of the following AudioServer commands, it is handled by the AudioServer: (1) addStream()—Adds a new stream to the server; (2) removeStream—Removes a stream from the server. This removes all objects associated with the stream (StreamSenders and MediaSources); (3) reset()—Removes all Streams from the server.

The Stream object (FIG. 7) represents a media stream from a single media source to one or more audio clients. One embodiment allows multiple sources to be “mixed”, or combinations of sources such as audio and video components to be presented as a single stream. Like AudioServer, Stream objects have a public handleCommand() method and get-Info() method.

StreamSender objects are created by Stream objects, and represent a connection to a remote audio client (end point). StreamSender is derived from the AsyncIOMultiplexer class, and AsyncIOMultiplexer functions are used to send data to remote clients.

The Stream object requests a frame of data from the send queue, which either gets the next frame from the queue or reads a new frame from the Media Source. It then sends the

data by calling `SendFrame` in each `StreamSender`. The `Stream` object also contains logic for timing out disconnected or high-latency clients, so they do not block packet transmissions to the entire client group. The statistics module may provide additional statistical functionality and it may use a callback to the `Stream` object.

The `MediaSource` object in FIG. 9 illustrates example functions used for opening, closing, and reading different types of media, for retrieving the audio parameters of that media, and for positioning (seeking the media) in the `Stream`. Subclasses can be used to provide this functionality for multiple media types, such as MP3 files, M4A files, AAC audio stream, WMA files, OSS audio devices (`dev/dspX`) on Linux, and MP3 streams on HTTP servers (i.e., Internet radio). This may include an analog input channel (or microphone) using the DSP device. In the case of audio files, the files may reside on a mounted local file system or network share. See FIG. 8. Audio Client

The design of an embodiment of the audio client will now be described in more detail. Like the audio server, the client can be a daemon process that is started automatically at system startup and runs on any client computing device or home automation device that has an audio playback device.

The audio client can run on one or more platforms like a media controller device or home theater controller device, or on a dedicated network audio I/O device like a speaker device. The design is modular, so that devices with multiple audio channels can run the audio client as a single process or as individual processes for each channel. Referring to FIG. 10, `AudioClient` may be the first object instantiated on startup, and creates an `ASCPAcceptor` object to listen for incoming connections on the client control port (e.g., 6100). As with the server, an incoming connection to `ASCPAcceptor` creates an `ASCPHandler` object which parses the incoming command and calls `handleCommand()` in `AudioClient`. `AudioClient` may then create a `Channel` object for each audio channel available on the device.

The `Channel` object handles the per-channel client commands passed to it by `AudioClient`, and may create several more per-channel objects, for example:

1. A `StreamAcceptor` to listen for incoming ASP audio stream data.
2. An `AudioBuffer` object for stream data.
3. An `AnnounceAcceptor` to listen for incoming ASP announcement data.
4. An `AudioBuffer` object for announcement data.
5. An `AudioDevice` object which provides an interface to the audio device driver.

`AudioClient` Object

The `AudioClient` in FIG. 10 may be the top-level object in the audio client process. After creating the `ASCPAcceptor` object and initializing one or more channels, the `run()` method is called, which calls the global `AsyncIOMultiplexer` `loop()` function and starts the audio client. The only command handled by `AudioClient` is `getInfo()`, which returns `getInfo()` for each channel in the channel list. All other commands are passed to the appropriate `Channel` object.

`Channel` Objects

Each channel object can represent a physical audio device, and holds `Acceptor` and `AudioBuffer` objects for stream and announcement data. Channel objects implement the `handleCommand()` and `getInfo()` methods. One of the key functions of the `Channel` object is to provide persistence of the `AudioDevice` and `AudioBuffer` objects when the `StreamReceiver` or `AnnounceReceiver` connections are closed or changed. The `Channel` objects can exist for the life of the `AudioClient`, and each incoming stream or announcement

connection closes any existing connection and becomes the “owner” of the channel object. This allows the connection to be switched seamlessly as long as there is enough data in the `AudioBuffer` object.

Streams and Receiver Objects

When a connection is made to the stream or announcement ports, the `StreamAcceptor` and `AnnounceAcceptor` objects create a `StreamReceiver` and `AnnounceReceiver` object, respectively as in FIG. 10. These objects are subclasses of the `ASPRceiver` class, which is an `AsyncIOMultiplexer` object that receives ASP packets and parses ASP headers. `StreamReceiver` Object

The `StreamReceiver` object can receive incoming ASP packets for the main audio stream. `StreamReceiver` may then instantiate a `Decoder` object for the ASP Frame Type (if it is not already instantiated), and passes the frame(s) to the `Decoder`'s decoding method. The PCM output is then added to the stream's `AudioBuffer` object using the `addData()` method, which queues it for transfer to the `AudioDevice`. If a non-zero drift factor has been calculated due to growing or shrinking of the audio buffer, the output from the decoder is first expanded or contracted by adding or removing samples according to a drift correction algorithm based on the sample rate. Data is then automatically transferred to the audio device when the audio device is able to accept another frame of data. Depending on certain flags in the ASP header, the behavior of the `StreamReceiver` may be altered as follows:

1. The decoder may be reset if the playnow flag is set or when a client is joining a session in progress.
2. If the null flag is set, the `Audio Buffer` is cleared and this stops playback quickly (i.e., as soon as the data in the kernel buffer is played out).

The `AnnounceReceiver` object can receive incoming ASP packets on the announcement socket. Like `StreamReceiver`, a `Decoder` object is instantiated as necessary, and the frames are decoded and sent to the announcement `AudioBuffer` using `addData()`. The first frame of a new announcement is put in `AudioBuffer` using `addDataAt()`, which places the first frame of the new announcement at the soonest possible play position. Unlike `StreamReceiver`, no drift correction is applied. The stream is terminated by closing the announcement connection.

When the mixing code in the `AudioDevice` object can find data in the announcement buffer, the `AudioDevice` object executes the mixing code, ducking the data being transferred from the main stream buffer. Correspondingly, when the announcement socket is closed and the announcement buffer is emptied, the ducking is stopped.

If an announcement connection is made while a previous announcement is still playing, the announcement is added to the queue using `addDataAt()`, clearing any old announcement data commencing at that time position.

The `AudioDevice` (see FIG. 10) is a class for interfacing to the DSP audio device on the client system, including setting the codec sample rate and controlling the volume. `AudioDevice` objects are a subclass of `AsyncIOMultiplexer`, using the file descriptor of the audio driver instead of a socket in the select set. Calling a `write select()` on the audio device will return ready when at least one complete fragment of audio data can be written to the device. This makes it convenient to implement `AudioDevice` as an `AsyncIOMultiplexer` and override the `handleWrite()` method to perform the transfer of data from the stream and announcement `AudioBuffer` objects to the audio device.

US 8,375,137 B2

11

When no data is available in either of the AudioBuffer objects, the transfer routine writes silent data to the audio device. This keeps the device playing at all times, if desired, and eliminates the clicks and pops that occur when the device is started and stopped.

The transfer routine in AudioDevice contains code for mixing announcement data with stream data (and possibly re-sampling the announcement data) at the time each fragment is transferred to the device. Since audio quality is not as important for announcements, the mixing algorithm will be chosen for high efficiency, and may be as simple as sample summing with the ducked stream data. The function of the mixer is controlled by the presence or absence of data in the announcement AudioBuffer.

Audio Buffering and AudioBuffer Class

Buffering is valuable in a streaming audio system to prevent dropouts in the audio output when the network connection experiences temporary pauses, or when other software running on either the server or the client performs large amounts of I/O, momentarily taking up system resources and causing jitter in the delivery of the audio stream.

In an embodiment, the audio client makes use of two buffers for audio data: 1) The kernel-level device buffer, and 2) the user-level main buffer. In the past, audio clients have used a single, large device buffer, which eliminated the allocation and management of a user-level buffer, but resulted in large delays when the stream was started or stopped, as the buffer always had to be filled or emptied autonomously by the kernel. The use of a single, large device buffer also presented problems when the sample rate of the data was changed between songs, since the audio client had to estimate when the first sample at the new sample rate was being played, and switch the codec sample rate at that time.

In the present technology, the kernel buffer is kept as small as practically possible, so that the response to play and stop commands is more immediate. The buffer can be large enough to prevent under-runs during heavy I/O (disk, frame buffer, etc.) in the kernel.

The user-level main buffer can be where incoming data is stored in an embodiment, before being passed to the kernel buffer. In previous systems, the synchronization scheme has relied on counting the number of fragments in the kernel buffer, and each incoming MP3 frame was immediately decoded and passed to the buffer. This tied the kernel fragment size to the size of an MP3 frame (4608 bytes), which was inefficient since two pages of memory were required for each frame. The improved main buffer design can allow the kernel fragment size to be set to any multiple of the page size, since an arbitrary number of bytes from the main buffer can be transferred to the kernel buffer. Also, since the transfer of frames into the main buffer is separate from the transfer of bytes from the main buffer to the kernel buffer, frames of arbitrary length (i.e. AAC) can be handled by the client, and any number of frames can be sent at the same time (in a single ASP packet), which significantly reduces the network overhead for formats with short frame lengths.

While the term home automation is used herein, these electronic components that have been described can be used with the central controller in other settings, including business, education, government, hotels, churches, broadcasting and entertainment facilities.

It is to be understood that the above-referenced arrangements are only illustrative of the application for the principles of the present invention. Numerous modifications and alternative arrangements can be devised without departing from the spirit and scope of the present invention. While the present invention has been shown in the drawings and fully described

12

above with particularity and detail in connection with what is presently deemed to be the most practical and preferred embodiment(s) of the invention, it will be apparent to those of ordinary skill in the art that numerous modifications can be made without departing from the principles and concepts of the invention as set forth herein.

The invention claimed is:

1. A streaming media system configured to prepare the streaming media for initial presentation, comprising:

a media server configured to send out media packets for a media stream at periodic clocked intervals;

a framer located with the media server to divide frames of the media stream into media packets;

a media client configured to receive the media packets for the media stream from the media server;

a send queue in the server, the send queue being configured to store a defined length of programming from the media stream, wherein the send queue can immediately fill a client's user buffer when an activation event occurs by transferring multiple frames in a single packet.

2. The system as in claim 1, wherein the activation event in the client device is a startup event or a reset event.

3. The system as in claim 1, further comprising a garbage collector configured to erase an unusable media packet in the send queue.

4. The system as in claim 1, wherein the garbage collector marks unusable media packets to indicate that the media packets will be overwritten.

5. The system as in claim 1, wherein the framer receives the media stream from a media source.

6. The system as in claim 1, wherein the media server sends media packets to a plurality of media clients.

7. The system as in claim 1, wherein the media client comprises a plurality of media clients which present the media stream to an end user across a plurality of rooms without noticeable delay between the plurality of rooms.

8. The system as in claim 1, wherein the periodic clocked intervals are set such that the media server transmits the media packets to the media client at an expected speed of media playback.

9. The system as in claim 1, wherein the media server communicates with the media client over a wired network.

10. The system as in claim 1, wherein the media server communicates with the media client over a wireless network.

11. The system as in claim 1, wherein the media server communicates with the media client using an audio stream control protocol.

12. A method of preparing an audio streaming system for initial presentation of a media stream, comprising:

loading a send queue in a media server with a defined length of programming;

receiving an activation event at a media server from a client device;

sending the defined length of programming in the send queue to the client device in an asynchronous manner without a clocking delay to allow the client device to startup quickly without waiting for a series of packets to be sent through clocking cycles;

receiving the defined length of programming into a user queue in the client device; and

playing the programming from the user queue in a media output device accessible to an end user.

13. The method as in claim 12, wherein the activation event in the client device is a startup event or a reset event.

14. The method as in claim 12, further comprising erasing unusable programming in the send queue, when the unusable programming is identified by a garbage collector.

US 8,375,137 B2

13

15. The method as in claim 12, further comprising overwriting expired programming in the send queue, wherein a garbage collector has marked the programming.

16. The method as in claim 12, wherein sending the defined length of programming comprises sending the defined length of programming at periodic clocked intervals, wherein the periodic clocked interval is set so that the media server transmits the media packets to the client device at an expected speed of media playback on the media output device. 5

17. The system as in claim 12, wherein the media server communicates with the media client over a wired network. 10

18. The method as in claim 12, wherein the media server communicates with the media client over a wireless network.

19. The method as in claim 12, wherein the media server communicates with the client device using an audio stream control protocol.

14

20. A streaming media system configured to prepare the streaming media for initial presentation, comprising:

a media server configured to send out media packets for a media stream at periodic clocked intervals;

a framer located with the media server to divide the media stream into media packets;

a media client configured to receive the media packets for the media stream from the media server;

a send queue in the server, the send queue being configured to store a defined length of programming from the media stream, wherein the send queue can fill a client's user buffer in an asynchronous manner without a clocking delay when an activation event occurs.

* * * * *

UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

PATENT NO. : 8,375,137 B2
APPLICATION NO. : 12/507754
DATED : February 12, 2013
INVENTOR(S) : Timothy D. Bowman

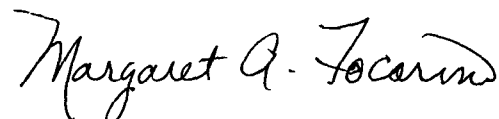
Page 1 of 1

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

In the Specification

In column 6, line 11 please delete “to ts_1 ” and replace it with --to ts_n --.

Signed and Sealed this
Twenty-sixth Day of November, 2013



Margaret A. Focarino
Commissioner for Patents of the United States Patent and Trademark Office

Exhibit C



(12) **United States Patent**
Milligan

(10) **Patent No.:** **US 11,140,426 B2**
(45) **Date of Patent:** **Oct. 5, 2021**

(54) **STREAMING MEDIA MULTIPLEXING WITH A MEDIA PROXY**

(71) Applicant: **Control4 Corporation**, Salt Lake City, UT (US)

(72) Inventor: **Thomas Allen Milligan**, Herriman, UT (US)

(73) Assignee: **Snap One, LLC**, Salt Lake City, UT (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 39 days.

(21) Appl. No.: **16/508,780**

(22) Filed: **Jul. 11, 2019**

(65) **Prior Publication Data**

US 2020/0021861 A1 Jan. 16, 2020

Related U.S. Application Data

(60) Provisional application No. 62/697,528, filed on Jul. 13, 2018.

(51) **Int. Cl.**

H04W 4/10 (2009.01)
H04N 21/2368 (2011.01)
H04L 29/06 (2006.01)

(52) **U.S. Cl.**

CPC **H04N 21/2368** (2013.01); **H04L 65/105** (2013.01); **H04L 65/1006** (2013.01); **H04L 65/601** (2013.01)

(58) **Field of Classification Search**

None
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

8,208,413 B1 *	6/2012	Bienn	H04Q 3/005
			370/271
9,674,140 B1 *	6/2017	Ivov	H04L 67/42
2004/0059942 A1 *	3/2004	Xie	H04L 61/2521
			726/12
2004/0131042 A1 *	7/2004	Lillie	H04L 12/1822
			370/351
2007/0019631 A1 *	1/2007	Jang	H04L 29/06027
			370/352
2009/0238168 A1 *	9/2009	Lavoie	H04L 65/1006
			370/352
2015/0223031 A1 *	8/2015	Denman	H04W 4/10
			370/328
2016/0380966 A1 *	12/2016	Gunnalan	H04L 61/2589
			709/226
2017/0238152 A1 *	8/2017	Patel	H04L 65/1006
			370/312
2017/0264929 A1 *	9/2017	Ahn	H04N 21/23608
2018/0054468 A1 *	2/2018	Whynot	H04L 65/105

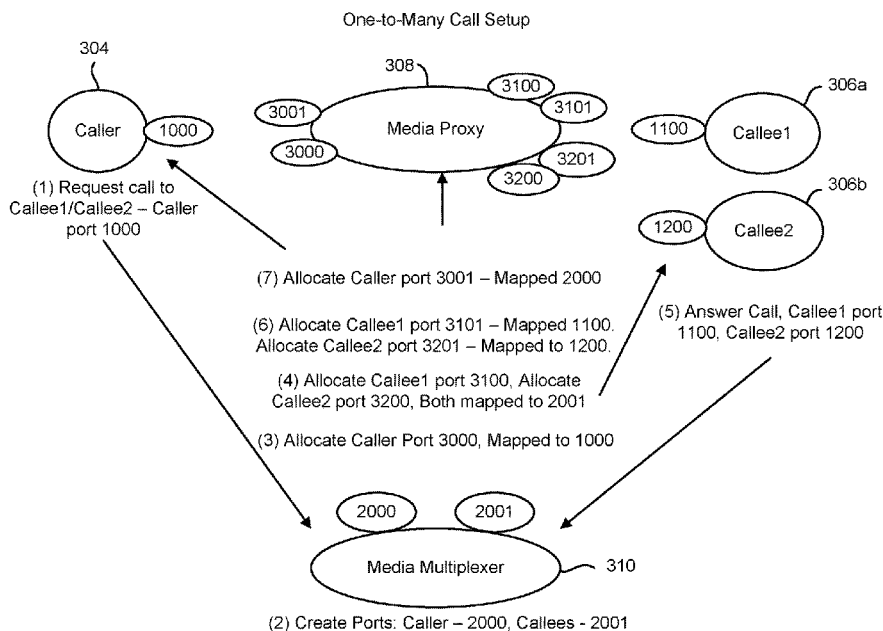
* cited by examiner

Primary Examiner — Sithu Ko
(74) Attorney, Agent, or Firm — Austin Rapp

(57) **ABSTRACT**

A method for communicating from one device to a plurality of receiving devices using a media multiplexer with a media proxy is described. A request for a call by the one device is received at the media multiplexer. Communication channels are allocated on the media proxy by the media multiplexer. A communication channel is allocated for each of the plurality of receiving devices. Media from the one device is received by the media multiplexer. The received media is sent to each of the plurality of receiving devices using the allocated communication channels.

8 Claims, 7 Drawing Sheets



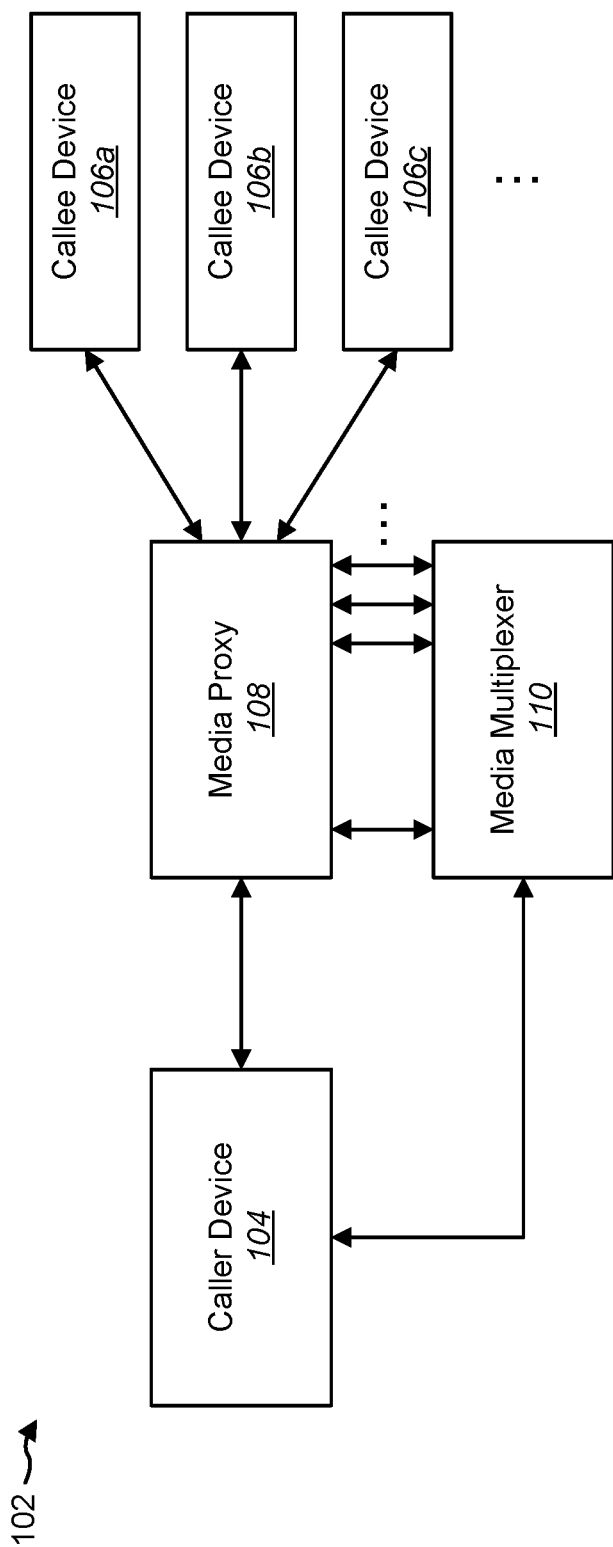


FIG. 1

200 ~>

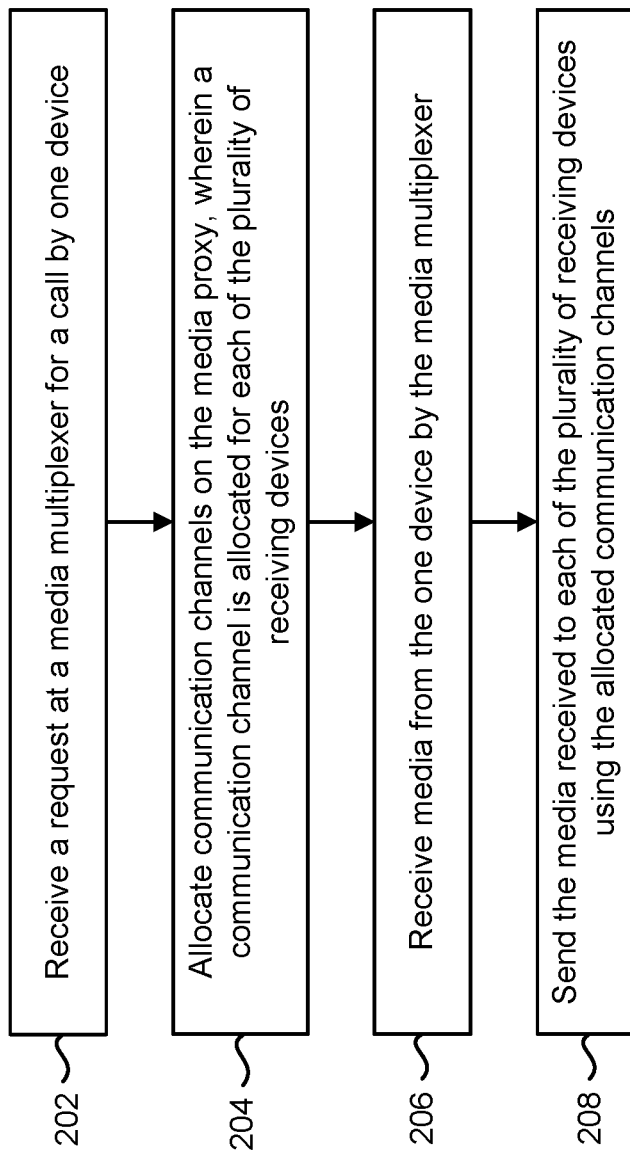


FIG. 2

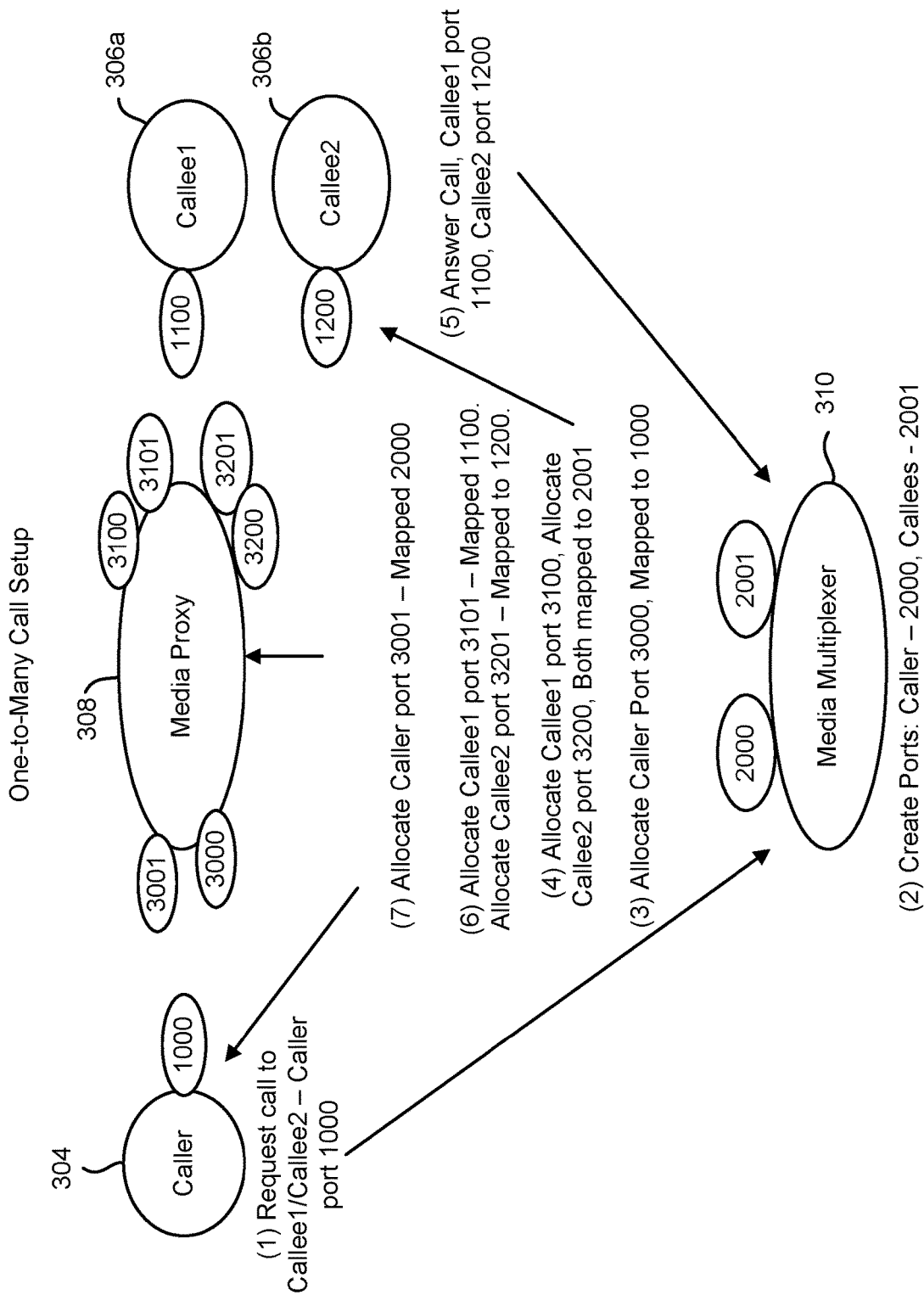


FIG. 3

One-to-Many Media Data Flow

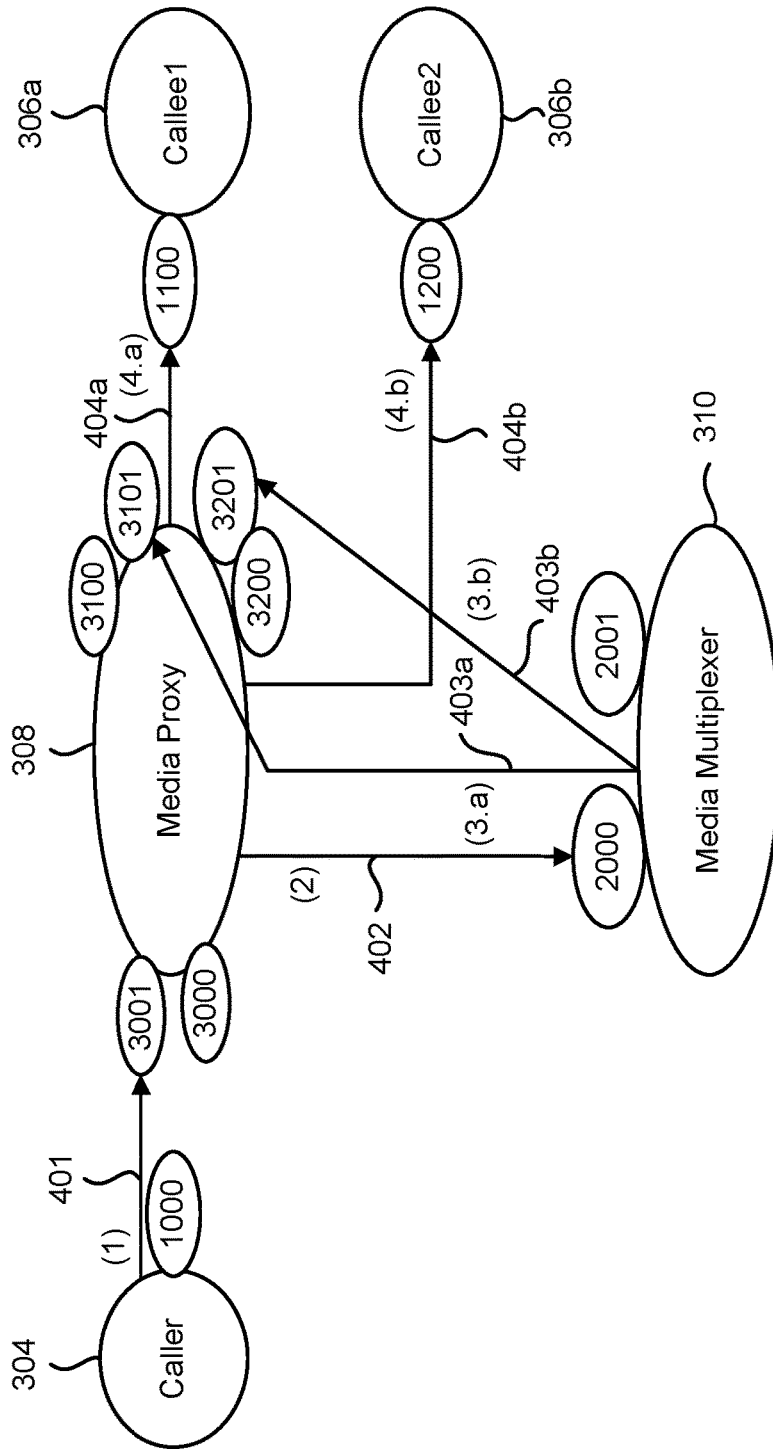


FIG. 4

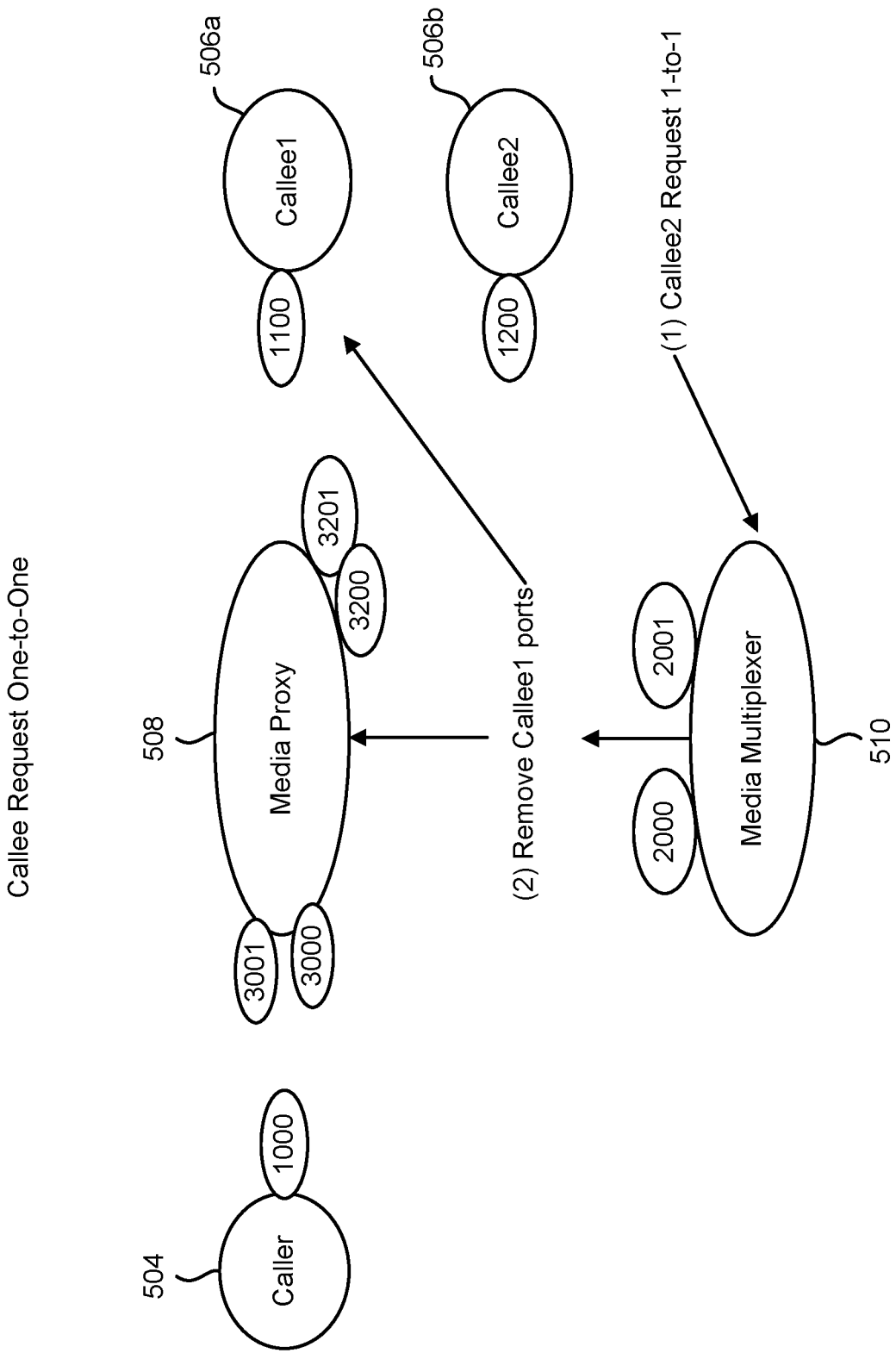


FIG. 5

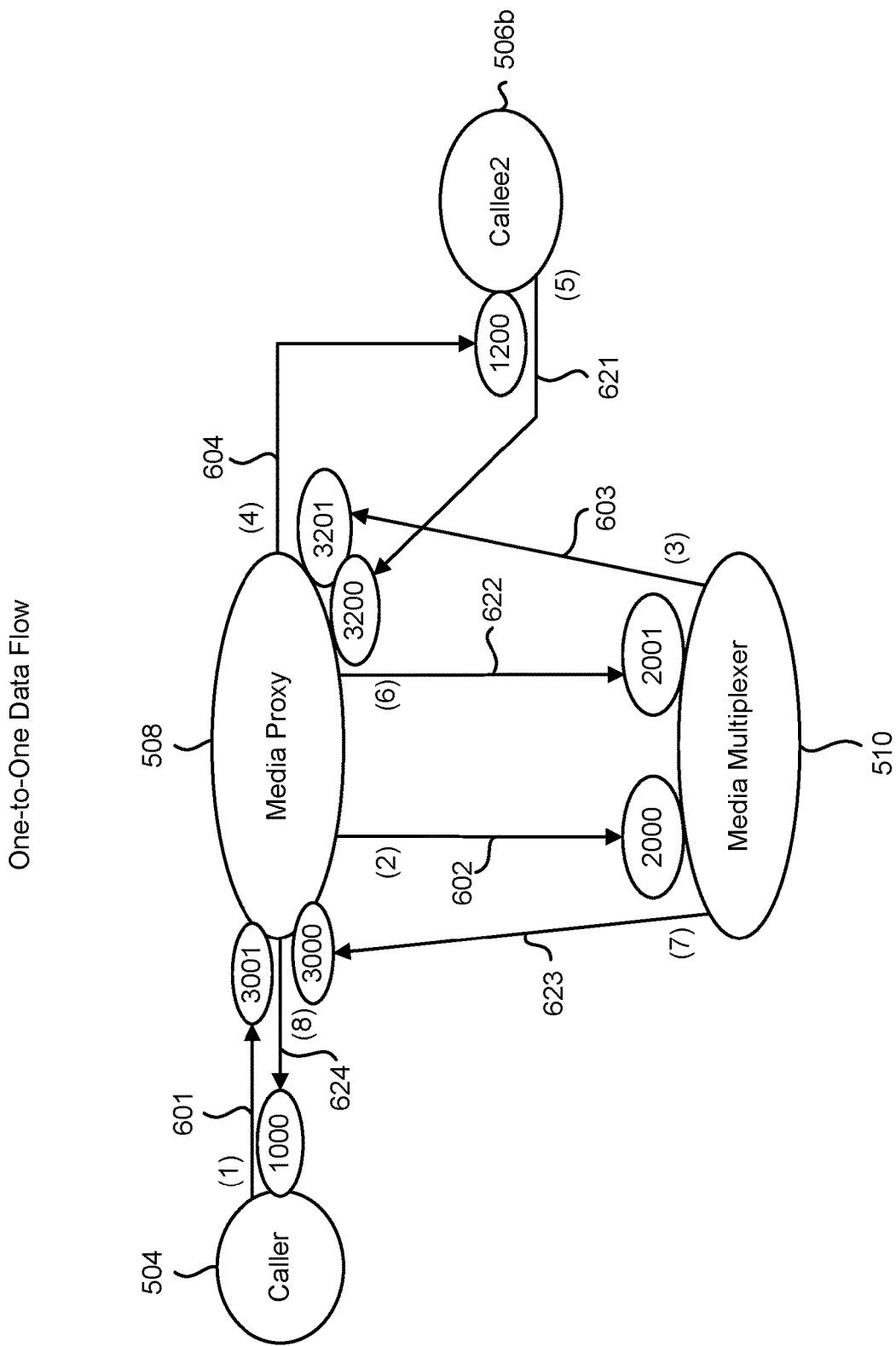


FIG. 6

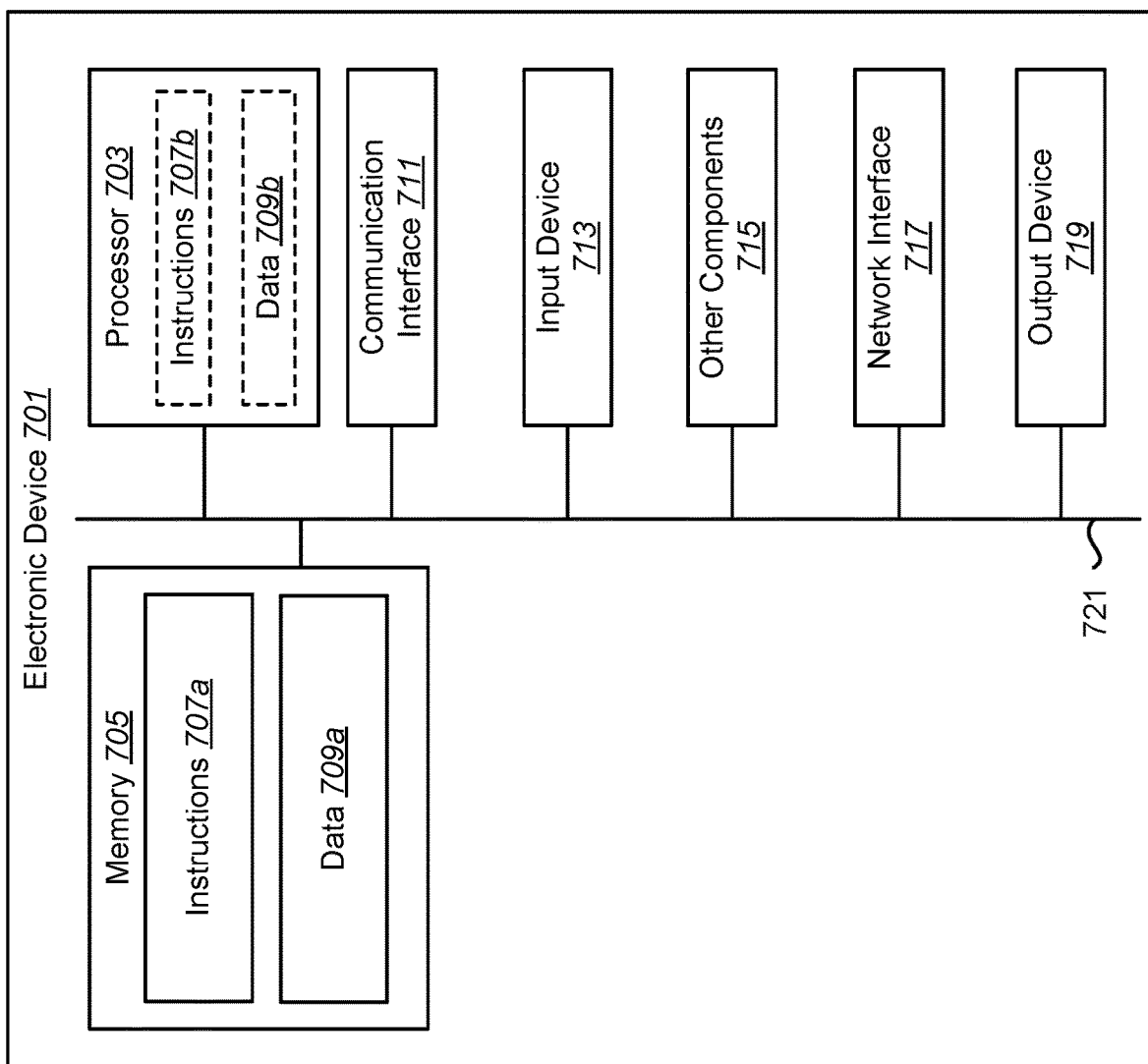


FIG. 7

US 11,140,426 B2

1

**STREAMING MEDIA MULTIPLEXING WITH
A MEDIA PROXY**

RELATED APPLICATIONS

This application is related to and claims priority from U.S. Provisional Patent Application No. 62/697,528, filed Jul. 13, 2018, for "STREAMING MEDIA MULTIPLEXING WITH A MEDIA PROXY," which is hereby incorporated by reference in its entirety.

TECHNICAL FIELD

The present disclosure relates generally to electronic devices. More specifically, the present disclosure relates to systems and methods for streaming media distributed by a media multiplexer with a media proxy to multiple receivers.

BACKGROUND

In recent years, the price of electronic devices has decreased dramatically. In addition, the types of electronic devices that can be purchased have continued to increase. For example, laptop computers, tablets, smart phones, DVD players, large screen TVs, multi-carousel CD and DVD players, MP3 players, video game consoles and similar consumer electronic items have become more widely available while continuing to drop in price.

The decreasing prices and increasing types of electronic components have packed today's homes and businesses with modern conveniences. Typical homes and businesses now include more electronic devices than ever before. While these electronic devices may provide convenience and entertainment, many also require control. Moreover, these electronic devices consume electrical power and may consume other resources.

Many electronic devices today have the ability to communicate with other electronic devices over different kinds of networks. In certain circumstances it is desirable to communicate from one device to multiple devices.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of a system for communicating from a caller device to a plurality of callee devices using a media proxy and a media multiplexer;

FIG. 2 is a flow diagram illustrating one configuration of a method for communicating from one device to a plurality of receiving devices using a media proxy and a media multiplexer;

FIG. 3 is a block diagram illustrating one configuration of a system for a one-to-many call setup;

FIG. 4 is a block diagram illustrating one configuration of a media data flow for the one-to-many call discussed in relation to FIG. 3;

FIG. 5 is a block diagram illustrating one configuration of a system for a one-to-one call setup;

FIG. 6 is a block diagram illustrating one configuration of the media data flow for the one-to-one call discussed in relation to FIG. 5; and

FIG. 7 is a block diagram illustrating various components that may be utilized in an electronic device.

DETAILED DESCRIPTION

A method for communicating from one device to a plurality of receiving devices using a media proxy is

2

described. The method includes receiving a request at a media multiplexer for a call by the one device. The method also includes allocating communication channels on the media proxy. A communication channel is allocated for each of the plurality of receiving devices. The method further includes receiving media from the one device by the media multiplexer. The method additionally includes sending the media received to each of the plurality of receiving devices using the allocated communication channels.

The method may also include receiving callee media from the plurality of receiving devices via the allocated communication channels by the media multiplexer. The callee media may be sent to the one device by the media multiplexer using the media proxy.

The method may also include receiving at the media multiplexer a request for exclusive streaming by a previous receiving device of the plurality of receiving devices. All of the allocated communication channels may be closed except for the communication channel being used by the previous receiving device.

A media multiplexer for communicating from one device to a plurality of receiving devices using a media proxy is also described. The media multiplexer includes a processor, a memory in electronic communication with the processor, and instructions stored in the memory. The instructions are executable by the processor to receive a request at the media multiplexer for a call by the one device. The instructions are also executable to allocate communication channels on the media proxy. A communication channel is allocated for each of the plurality of receiving devices. The instructions are further executable to receive media from the one device by the media multiplexer. The instructions are additionally executable to send the media received to each of the plurality of receiving devices using the allocated communication channels.

Another method for communicating from one device to a plurality of receiving devices using a media proxy is described. The method includes receiving a request for a one-to-many call from a caller device to a plurality of callee devices. The method also includes creating a caller port by the media multiplexer to receive media from the caller device. The method further includes creating a receiving port to receive media from the plurality of callee devices. The method additionally includes allocating a port by the media multiplexer on a media proxy for sending media to the caller device. The method also includes allocating a port by the media multiplexer on the media proxy for each of the plurality of callee devices that directs media to the media multiplexer. The method further includes forwarding the request to each of the plurality of callee devices using a unique port for each callee device. The method additionally includes answering the call by each of the plurality of callee devices using the port for receiving media on. The method also includes allocating a port by the media multiplexer on a media proxy for each callee answer. The port allocated is used to forward media to the associated callee device. The method further includes allocating a port by the media multiplexer on the media proxy. The port allocated directs caller media to the media multiplexer. The method additionally includes notifying the caller device of the port on the media proxy to send media to the media multiplexer.

Another media multiplexer for communicating from one device to a plurality of receiving devices using a media proxy is described. The media multiplexer receives a request for a one-to-many call from a caller device to a plurality of callee devices. The media multiplexer creates a caller port to receive media from the caller device. The media multiplexer

US 11,140,426 B2

3

creates a receiving port to receive media from the plurality of callee devices. The media multiplexer allocates a port by the media multiplexer on a media proxy for sending media to the caller device. The media multiplexer allocates a port by the media multiplexer on the media proxy for each of the plurality of callee devices that directs media to the media multiplexer. The media multiplexer forwards the request to each of the plurality of callee devices using a unique port for each callee. The media multiplexer answers the call by each of the plurality of callee devices using the port for receiving media on. The media multiplexer allocates a port by the media multiplexer on a media proxy for each callee answer. The port allocated is used to forward media to the associated callee device. The media multiplexer allocates a port by the media multiplexer on the media proxy. The port allocated directs caller media to the media multiplexer. The media multiplexer notifies the caller device of the port on the media proxy to send media to the media multiplexer.

Sending a media stream to an endpoint that is behind a firewall can be facilitated with a media proxy. Media proxies provide public network access ports where media can be sent and then forwarded by the media proxy to the intended recipient. If the stream is intended for multiple recipients, then a media multiplexer may be used in conjunction with a media proxy to forward the media. To support multiple receivers, when a media proxy receives media, the media is forwarded to a media multiplexer, which subsequently forwards the media back to multiple media proxy ports for distribution to the group of intended recipients. The present systems and methods describe the process and procedures that allow a media proxy and media multiplexer to collaboratively distribute a media stream to a group of recipients.

One possible use case is the scenario where there is a camera by the front door of a home that is activated whenever someone rings the doorbell. When someone rings the doorbell, media is sent out, but it needs to be sent to all devices in the group that is associated with the doorbell. The devices in the group may be at various locations depending on where the users of the devices are located. For example, the devices could be at home, or at work, travelling, etc. With the present systems and methods, all devices that are associated with the front door doorbell group may see the doorbell alert and image and/or video. A method is also provided where one specific device can answer the door.

The following terms are used in the present description. Definitions for how these terms are used herein are provided as follows:

Port—Network communication port that is identified by network host address and a port number.

Network Service—A process that runs on a public network host that provides a network application programming interface (API) using network communications.

Media Stream—audio, video, or other media formats that change data over time that must be continually updated.

Media Proxy—Network service that provides public network ports where media can be sent and then forwarded to the intended recipient. The sender and receiver agree on public ports since they cannot always communicate directly.

Media Multiplexer—Network service that maps a single media stream source to multiple target destinations.

SIP—Session Initiation Protocol, which may be used for signaling and controlling multimedia communication sessions.

SDP—Session Description Protocol, which may be used to define the media formats, media attributes, and media port locations.

4

Various configurations are now described with reference to the figures, where like reference numbers may indicate functionally similar elements. The systems and methods as generally described and illustrated in the figures herein could be arranged and designed in a wide variety of different configurations. Thus, the following more detailed description of several configurations, as represented in the Figures, is not intended to limit scope, as claimed, but is merely representative of the systems and methods. As used herein, the term “plurality” may indicate two or more. For example, a plurality of components may refer to two or more components.

FIG. 1 is a block diagram of a system **102** for communicating from a caller device **104** to a plurality of callee devices **106a**, **106b**, **106c** using a media proxy **108** and a media multiplexer **110**. The caller device **104**, callee devices **106a-c**, media proxy **108** and media multiplexer **110** may be electronic devices. Examples of electronic devices include home controllers, audio/video receivers, servers, computers (e.g., desktop computers, laptop computers, etc.), network devices, gaming consoles, smart televisions, smartphones, tablet devices, vehicles, automobiles, aircraft, appliances, etc. In some configurations, the electronic device may be located in a building, home, business, vehicle, etc., and/or may be integrated into one or more devices (e.g., vehicles, mobile devices, etc.). The electronic device may include one or more components or elements. One or more of the components or elements may be implemented in hardware (e.g., circuitry), a combination of hardware and software (e.g., a processor with instructions), and/or a combination of hardware and firmware.

In some configurations, an electronic device (e.g., caller device **104**, callee devices **106**, media proxy **108** and/or media multiplexer **110**) may include a processor, a memory, and/or one or more communication interfaces. The processor may be coupled to (e.g., in electronic communication with) the memory and/or communication interface(s).

In some configurations, an electronic device (e.g., caller device **104**, callee devices **106**, media proxy **108** and/or media multiplexer **110**) may be configured to perform one or more of the functions, procedures, methods, steps, etc., described in connection with one or more of FIGS. 1-7. Additionally or alternatively, an electronic device may include one or more of the structures described in connection with one or more of FIGS. 1-7.

The media multiplexer **110** may perform call management by setting up calls, receiving data and forwarding the data where it needs to go. In some implementations, the media multiplexer **110** may receive a request for a call from the caller device **104**. For example, the call may indicate media that is to be communicated with a number of callee devices **106a-c**. In some cases, the call may be sent to establish a media stream from the caller device **104** to the callee devices **106a-c**.

Upon receiving the call, the media multiplexer **110** may allocate communication channels on the media proxy **108**. As used herein, a communication channel may be a port used to communicate with an electronic device over a network. The media multiplexer **110** may allocate a communication channel on the media proxy **108** for each of the callee devices **106a-c**. For example, a first communication channel may facilitate communication between a first callee device **106a** and the media proxy **108**, a second communication channel may facilitate communication between a second callee device **106b** and the media proxy **108**, and so forth.

US 11,140,426 B2

5

The media multiplexer **110** may receive media (e.g., a media stream) from the caller device **104**. For example, the caller device **104** may send the media to the media multiplexer **110**. In another example, the caller device **104** may send the media to the media proxy **108**, which forwards the media to the media multiplexer **110**.

Upon receiving the media, the media multiplexer **110** may send the media to each of the callee devices **106a-c** (receiving devices) using the communication channels allocated on the media proxy **108**. For example, the media multiplexer **110** may send the media to the first callee device **106a** using the first communication channel on the media proxy **108**. The media multiplexer **110** may send the media to the second callee device **106b** using the second communication channel on the media proxy **108**, and so forth.

In some examples, media may also be received from the callee devices **106a-c** via the allocated communication channels on the media proxy **108**. For example, a callee device **106** may send media to the media proxy **108** via the allocated communication channel. The media proxy **108** may then forward the media from the callee device **106** to the media multiplexer **110**. The media multiplexer **110** may send the received media to the caller device **104** using the media proxy **108**.

In some cases, one of the callee devices **106a-c** may request exclusive streaming. When exclusive streaming is requested the media multiplexer **110** may close all of the allocated communication channels except for the communication channel being used by the callee device **106** that requested exclusive streaming.

In some examples, the media proxy **108** and media multiplexer **110** may both be cloud services. For instance, the media proxy **108** and media multiplexer **110** may communicate with the caller device **104** and/or callee devices **106a-c** over the Internet.

FIG. 2 is a flow diagram **200** illustrating one configuration of a method for communicating from one device (e.g., a caller device **104**) to a plurality of receiving devices (e.g., callee devices **106a-c**) using a media proxy **108** and a media multiplexer **110**. A request for a call, sent by the one device **104**, may be received **202** by a media multiplexer **110**. The call may indicate a number of receiving devices (e.g., callee devices **106a-c**) that are to receive the call.

The media multiplexer **110** may allocate **204** communication channels on the media proxy **108** for the plurality of receiving devices (e.g., callee devices **106a-c**). A communication channel may be allocated on the media proxy **108** for each of the plurality of receiving devices **106a-c**.

Media may be received **206** from the one device **104** by the media multiplexer **110**. For example, the one device **104** may send the media to the media multiplexer **110**. In another example, the one device **104** may send the media to the media proxy **108**, which forwards the media to the media multiplexer **110**.

The received media may be sent **208** to each of the plurality of receiving devices **106a-c** using the allocated communication channels. For example, the media multiplexer **110** may send **208** the media to the first receiving device **106a** using the first communication channel on the media proxy **108**. The media multiplexer **110** may send the media to the second receiving device **106b** using the second communication channel on the media proxy **108**, and so forth.

In some examples, the media multiplexer **110** may receive callee media from the plurality of receiving devices via the allocated communication channels by the media multiplexer.

6

The media multiplexer **110** may send the callee media to the one device using the media proxy **108**.

In some examples, the media multiplexer **110** may receive a request for exclusive streaming by a previous receiving device of the plurality of receiving devices. The media multiplexer **110** may close all of the allocated communication channels except for the communication channel being used by the previous receiving device.

FIG. 3 is a block diagram illustrating one configuration of a system for a one-to-many call setup. In this example, a call may be set up between a single caller device **304** (e.g., Caller) and multiple callee devices **306a-b** (e.g., Callee1, Callee2, etc.).

In a first step (1), the caller device **304** may request a one-to-many call and specifies the caller port **1000** to receive the media. For example, the caller device **304** may send the call request to the media multiplexer **310**. In this step, the caller device **304** may communicate the call request and caller port **1000** to the media multiplexer **310**.

In a second step (2), the media multiplexer **310** may create a port **2000** to receive media from the caller device **304**. The media multiplexer **310** may also create a port **2001** to receive media from the callee devices **306a-b**.

In a third step (3) the media multiplexer **310** may allocate a port **3000** on a media proxy **308** for sending media to the caller device **304**. In this case, the caller port **3000** of the media proxy **308** may be mapped to the caller port **1000** of the caller device **304**.

In a fourth step (4), the media multiplexer **310** may allocate a port on the media proxy **308** for each callee device **306** that directs media sent by a callee device **306** to the media multiplexer **310**. For example, the media multiplexer **310** may allocate a port **3100** on the media proxy **308** to forward media sent by a first callee device (Callee1) **306a** to the media multiplexer **310**. The port **3100** may be mapped to port **2001** on the media multiplexer **310**. The media multiplexer **310** may allocate a port **3200** on the media proxy **308** to forward media sent by a second callee device (Callee2) **306b** to the media multiplexer **310**. The port **3200** may also be mapped to port **2001** on the media multiplexer **310**.

The media multiplexer **310** may then forward the call request with the unique port (e.g., port **3100** and **3200**) for each callee device **306**. For example, the media multiplexer **310** may forward the call request to the first callee device (Callee1) **306a** that includes information about port **3100** on the media proxy **308**. The media multiplexer **310** may also forward the call request to the second callee device (Callee2) **306b** that includes information about port **3200** on the media proxy **308**.

In a fifth step (5), each callee **306** may answer the call with the port that they can receive media on. For example, the first callee device (Callee1) **306a** may answer the call by responding to the media multiplexer **310** with information about port **1100**. In this case, port **1100** may be used by the media proxy **308** to send media to the first callee device (Callee1) **306a**. The second callee device (Callee2) **306b** may answer the call by responding to the media multiplexer **310** with information about port **1200**. In this case, port **1200** may be used by the media proxy **308** to send media to the second callee device (Callee2) **306b**.

In a sixth step (6), for each callee device **306** that answers, the media multiplexer **310** may allocate a port on the media proxy **308** that is used to forward media to the associated callee **306**. For example, the media multiplexer **310** may allocate Callee1 port **3101** on the media proxy **308** that is mapped to port **1100** on the first callee device (Callee1) **306a**. The media multiplexer **310** may allocate Callee2 port

3201 on the media proxy **308** that is mapped to port **1200** on the second callee device (Callee2) **306b**.

In a seventh step (7), the media multiplexer **310** may allocate a port **3001** on the media proxy **308** that directs caller media to the media multiplexer **310**. The port **3001** on the media proxy **308** may be mapped to port **2000** on the media multiplexer **310**. The media multiplexer **310** may then notify the caller **304** of the port **3001** to send media to.

In some approaches, the signaling to set up the media streams may be implemented using Session Initiation Protocol (SIP), which may use Session Description Protocol (SDP). Using SIP with SDP, the devices (e.g., caller device **304**, callee devices **306a-b**, the media multiplexer **310** and/or media proxy **308**) may negotiate how to communicate between the source and the client's endpoints.

FIG. 4 is a block diagram illustrating one configuration of a media data flow for the one-to-many call discussed in relation to FIG. 3. In a first step (1), the media data starts at the caller device **304** and is sent **401** to the media proxy **308**. For example, the caller device **304** may send media (e.g., a media stream) to port **3001** of the media proxy **308**.

In a second step (2), the media data may be sent **402** to the media multiplexer **310**. For example, because port **3001** of the media proxy **308** is mapped to port **2000** of the media multiplexer **310**, the media proxy **308** may forward media received on port **3001** to port **2000** of the media multiplexer **310**.

In a third step, the media multiplexer **310** may send the media data on two streams to the media proxy **308**. For example, in step **3.a**, the media multiplexer **310** may send **403a** the media data on a first stream to port **3101** of the media proxy **308**. In step **3.b**, the media multiplexer **310** may send **403b** the media data on a second stream to port **3201** of the media proxy **308**.

In a fourth step, the media proxy **308** forwards the separate data streams to the callee devices **306a**, **306b**. For example, in step **4.a**, the media proxy **308** may send **404a** the media data to port **1100** of the first callee device (Callee1) **306a** because port **3101** of the media proxy **308** is mapped to port **1100** of the first callee device (Callee1) **306a**. In step **4.b**, the media proxy **308** may send **404b** the media data to port **1200** of the second callee device (Callee2) **306b** because port **3201** of the media proxy **308** is mapped to port **1200** of the second callee device (Callee2) **306b**.

It should be noted that the media data flow is shown in FIG. 4 along the flow from **401**, **402**, **403a** & **403b** and ending with **404a** & **404b**. In some implementations, the media multiplexer and/or the media proxy may handle a set of media streams as a single call and process the streams aggregately. For example, media streams originating from the caller device may be combined with media streams from the callee devices to form a single call.

In some implementations, the media proxy **308** may transcode the source media to different formats as requested by each recipient. For example, the media proxy **308** may transcode the media sent from the caller device **304** to a format that is compatible with a callee device **306**.

In some implementations, the media proxy **308** may decrypt the media from the source's security context and encrypt the media for each recipient based on the receivers' security context. For example, the media proxy **308** may decrypt the media from the security context of the caller device **304** and may encrypt the media for a security context of a callee device **306**.

FIG. 5 is a block diagram illustrating one configuration of a system for a one-to-one call setup. It should be noted that the ports described in FIG. 5 may be allocated as described

in FIG. 3. For example, a one-to-many call may be set up between the caller device **504** and the callee devices **506a-b** as described in FIG. 3. The port numbering described in FIG. 5 corresponds to the port numbering of FIG. 3.

In a first step (1) of a one-to-one call setup, after the one-to-many call is setup, a member of the callee group **506a**, **506b** requests exclusive one-to-one media streaming with the caller device **504**. In this example, the second callee device (Callee2) **506b** requests a one-to-one call. The second callee device (Callee2) **506b** may send the one-to-one call request to the media multiplexer **510**.

In a second step (2), the media multiplexer **510** may remove all media ports on the media proxy **508** for the other callee devices. The media multiplexer **510** may also notify the other callee devices that their calls have been terminated. For example, upon receiving the one-to-one call request from the second callee device (Callee2) **506b**, the media multiplexer **510** may remove port **3100** and port **3101** from the media proxy **508** for the first callee device (Callee1) **506a**. The media multiplexer **510** may then notify the first callee device (Callee1) **506a** that its call has been terminated.

FIG. 6 is a block diagram illustrating one configuration of the media data flow for the one-to-one call discussed in relation to FIG. 5. The media data flowing from the caller device (Caller) **504** to the callee device (Callee2) **506b** is shown along the pathway **601**, **602**, **603** and **604**. The media data flowing from the callee device **506b** to the caller **504** is shown along the pathway **621**, **622**, **623** and **624**.

In a first step (1), the media data intended for the callee device (Callee2) **506b** starts at the caller device **504** and is sent **601** to the media proxy **508**. For example, the caller device **504** may send media (e.g., a media stream) to port **3001** of the media proxy **508**.

In a second step (2), the media data may be sent **602** to the media multiplexer **510**. For example, because port **3001** of the media proxy **508** is mapped to port **2000** of the media multiplexer **510**, the media proxy **508** may forward media received on port **3001** to port **2000** of the media multiplexer **510**.

In a third step (3), the media multiplexer **510** may send the media data on a stream to the media proxy **508**. For example, the media multiplexer **510** may send **603** the media data on a stream to port **3201** of the media proxy **508**.

In a fourth step (4), the media proxy **508** may forward the data stream to the callee device **506b**. For example, the media proxy **508** may send **604** the media data to port **1200** of the callee device (Callee2) **506b** because port **3201** of the media proxy **508** is mapped to port **1200** of the callee device (Callee2) **506b**.

In a fifth step (5), the callee device (Callee2) **506b** may send **621** media to the media proxy **508**. For example, the callee device (Callee2) **506b** may send **621** the media data to port **3200** of the media proxy **508**. It should be noted that media sent from the callee device **506** to the caller device **504** may be referred to as "callee media."

In a sixth step (6), the media data may be sent **622** to the media multiplexer **510**. For example, because port **3200** of the media proxy **508** is mapped to port **2001** of the media multiplexer **510**, the media proxy **508** may forward media received on port **3200** to port **2001** of the media multiplexer **510**.

In a seventh step (7), the media multiplexer **510** may send the media data on a stream to the media proxy **508**. For example, the media multiplexer **510** may send **623** the media data on a stream to port **3000** of the media proxy **508**.

In an eighth step (8), the media proxy 508 forwards the data stream to the caller device 504. For example, the media proxy 508 may send 624 the media data to port 1000 of the caller device 504 because port 3000 of the media proxy 508 is mapped to port 1000 of the caller device 504.

In some implementations, the media proxy 508 may transcode the source media to different formats as requested by the callee device 506b. In some implementations, the media proxy 508 may decrypt the media from the security context of the caller device 504 and may encrypt the media for a security context of the callee device 506b.

The electronic devices herein may also be referred to as computing devices. Examples of computing devices include desktop computers, laptop computers, tablet devices, notebooks, cellular phones, smart phones, smart watches, routers, personal digital assistants (PDAs), thermostats, controllers, sensors, actuators, etc.

Various devices described herein may be configured to communicate with or be part of the cloud. As used herein, the term “cloud” refers to an Internet-based computing network of one or more remote servers.

FIG. 7 is a block diagram illustrating various components that may be utilized in an electronic device. Examples of electronic devices include a caller device, callee device, media proxy and media multiplexer as described herein.

An electronic device may include a broad range of digital computers, including microcontrollers, hand-held computers, personal computers, servers, mainframes, supercomputers, minicomputers, workstations and any variation or related device thereof. In some configurations, the electronic device may be an appliance. Additionally or alternatively, the electronic device may be an embedded device inside an otherwise complete device (e.g., within an appliance).

The electronic device 701 is shown with a processor 703 and memory 705. The processor 703 may control the operation of the electronic device 701 and may be embodied as a microprocessor, a microcontroller, a digital signal processor (DSP) or other device known in the art. The processor 703 typically performs logical and arithmetic operations based on program instructions 707a and/or data 709a stored within the memory 705. The instructions 707a in the memory 705 may be executable to implement the methods described herein. FIG. 8 illustrates instructions 707b and/or data 709b being loaded onto the processor 703. The instructions 707b and/or data 709b may be the instructions 707a and/or data 709a (or portions thereof) stored in memory 705.

The electronic device 701 may also include one or more communication interfaces 711 and/or network interfaces 717 for communicating with other electronic devices. The communication interface(s) 711 and the network interface(s) 717 may be based on wired communication technology and/or wireless communication technology, such as ZigBee®, WiMax®, WiFi®, Bluetooth® and/or cellular protocols, such as GSM®, etc.

The electronic device 701 may also include one or more input devices 713 and one or more output devices 719. The input devices 713 and output devices 719 may facilitate user input/user output. Other components 715 may also be provided as part of the electronic device 701.

Instructions 707a and data 709a may be stored in the memory 705. The processor 703 may load and execute instructions 707b from the instructions 707a in memory 705 to implement various functions. Executing the instructions 707a may involve the use of the data 709a that is stored in the memory 705. The instructions 707b and/or data 709b may be loaded onto the processor 703. The instructions 707 are executable to implement the one or more methods shown

herein and the data 709 may include one or more of the various pieces of data described herein.

The memory 705 may be any electronic component capable of storing electronic information. The memory 705 may be embodied as random access memory (RAM), read-only memory (ROM), magnetic disk storage media, optical storage media, flash memory devices in RAM, on-board memory included with the processor, EPROM memory, EEPROM memory, an ASIC (Application Specific Integrated Circuit), registers and so forth, including combinations thereof. The various components of the electronic device may be coupled together by a bus system 721, which may include a power bus, a control signal bus and a status signal bus, in addition to a data bus. However, for the sake of clarity, the various buses are illustrated in FIG. 7 as the bus system 721.

In the above description, reference numbers have sometimes been used in connection with various terms. Where a term is used in connection with a reference number, it may refer to a specific element that is shown in one or more of the Figures. Where a term is used without a reference number, it may refer generally to the term without limitation to any particular Figure.

The term “determining” encompasses a wide variety of actions and, therefore, “determining” can include calculating, computing, processing, deriving, investigating, looking up (e.g., looking up in a table, a database or another data structure), ascertaining and the like. Also, “determining” can include receiving (e.g., receiving information), accessing (e.g., accessing data in a memory) and the like. Also, “determining” can include resolving, selecting, choosing, establishing and the like.

The phrase “based on” does not mean “based only on,” unless expressly specified otherwise. In other words, the phrase “based on” describes both “based only on” and “based at least on.”

The term “processor” should be interpreted broadly to encompass a general purpose processor, a central processing unit (CPU), a microprocessor, a digital signal processor (DSP), a controller, a microcontroller, a state machine and so forth. Under some circumstances, a “processor” may refer to an application specific integrated circuit (ASIC), a programmable logic device (PLD), a field programmable gate array (FPGA), etc. The term “processor” may refer to a combination of processing devices e.g., a combination of a DSP and a microprocessor, a plurality of microprocessors, one or more microprocessors in conjunction with a DSP core, or any other such configuration.

The term “memory” should be interpreted broadly to encompass any electronic component capable of storing electronic information. The term memory may refer to various types of processor-readable media such as random access memory (RAM), read-only memory (ROM), non-volatile random access memory (NVRAM), programmable read-only memory (PROM), erasable programmable read-only memory (EPROM), electrically erasable PROM (EEPROM), flash memory, magnetic or optical data storage, registers, etc. Memory is said to be in electronic communication with a processor if the processor can read information from and/or write information to the memory. Memory that is integral to a processor is in electronic communication with the processor.

The terms “instructions” and “code” should be interpreted broadly to include any type of computer-readable or processor-readable statement(s). For example, the terms “instructions” and “code” may refer to one or more programs, routines, sub-routines, functions, procedures, etc.

US 11,140,426 B2

11

“Instructions” and “code” may comprise a single computer-readable statement or many computer-readable statements.

The term “computer-readable medium” refers to any available medium that can be accessed by a computer or processor. By way of example, and not limitation, a computer-readable medium may comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium that can be used to carry or store desired program code in the form of instructions or data structures and that can be accessed by a computer. A computer-readable medium may be tangible and non-transitory. Disk and disc, as used herein, includes compact disc (CD), laser disc, optical disc, digital versatile disc (DVD), floppy disk and Blu-ray® disc where disks usually reproduce data magnetically, while discs reproduce data optically with lasers.

Software or instructions may also be transmitted over a transmission medium. For example, if the software is transmitted from a website, server, or other remote source using a coaxial cable, fiber optic cable, twisted pair, digital subscriber line (DSL), or wireless technologies such as infrared, radio and microwave, then the coaxial cable, fiber optic cable, twisted pair, DSL, or wireless technologies such as infrared, radio and microwave are included in the definition of transmission medium.

The methods disclosed herein comprise one or more steps or actions for achieving the described method. The method steps and/or actions may be interchanged with one another without departing from the scope of the claims. In other words, unless a specific order of steps or actions is required for proper operation of the method that is being described, the order and/or use of specific steps and/or actions may be modified without departing from the scope of the claims.

It is to be understood that the claims are not limited to the precise configuration and components illustrated above. Various modifications, changes and variations may be made in the arrangement, operation and details of the systems, methods and apparatus described herein without departing from the scope of the claims.

What is claimed is:

1. A method for communicating from one device to a plurality of receiving devices using a media proxy, comprising:

receiving a request, sent from the one device, at a media multiplexer for a call by the one device;
 allocating communication channels on the media proxy, wherein a communication channel is allocated for each of the plurality of receiving devices;
 receiving media from the one device by the media multiplexer;
 sending the media received to each of the plurality of receiving devices using the allocated communication channels;
 receiving at the media multiplexer a request for exclusive streaming by a previous receiving device of the plurality of receiving devices; and
 closing all of the allocated communication channels except for the communication channel being used by the previous receiving device.

2. The method of claim 1, further comprising receiving callee media from the plurality of receiving devices via the allocated communication channels by the media multiplexer.

3. The method of claim 2, further comprising sending the callee media to the one device by the media multiplexer using the media proxy.

12

4. A media multiplexer for communicating from one device to a plurality of receiving devices using a media proxy, comprising:

a processor;
 a memory in electronic communication with the processor; and
 instructions stored in the memory, the instructions being executable by the processor to:
 receive a request, sent from the one device, at the media multiplexer for a call by the one device;
 allocate communication channels on the media proxy, wherein a communication channel is allocated for each of the plurality of receiving devices;
 receive media from the one device by the media multiplexer;
 send the media received to each of the plurality of receiving devices using the allocated communication channels;
 receive at the media multiplexer a request for exclusive streaming by a previous receiving device of the plurality of receiving devices; and
 close all of the allocated communication channels except for the communication channel being used by the previous receiving device.

5. The media multiplexer of claim 4, further comprising instructions to receive callee media from the plurality of receiving devices via the allocated communication channels by the media multiplexer.

6. The media multiplexer of claim 5, further comprising instructions to send the callee media to the one device by the media multiplexer using the media proxy.

7. A method for communicating from one device to a plurality of receiving devices using a media proxy, comprising:

receiving a request for a one-to-many call from a caller device to a plurality of callee devices;
 creating a caller port by a media multiplexer to receive media from the caller device;
 creating a receiving port to receive media from the plurality of callee devices;
 allocating a first port by the media multiplexer on a media proxy for sending media to the caller device;
 allocating a corresponding port by the media multiplexer on the media proxy for each of the plurality of callee devices that directs media to the media multiplexer;
 forwarding the request to each of the plurality of callee devices using the corresponding port for each callee device;
 answering the call by each of the plurality of callee devices using a port for receiving media on;
 allocating a respective port by the media multiplexer on the media proxy for each callee answer, wherein each respective port allocated is used to forward media to the associated callee device;
 allocating a second port by the media multiplexer on the media proxy, wherein the second port allocated directs caller media to the media multiplexer; and
 notifying the caller device of the second port on the media proxy to send media to the media multiplexer.

8. A media multiplexer for communicating from one device to a plurality of receiving devices using a media proxy, comprising:

a processor;
 a memory in electronic communication with the processor; and
 instructions stored in the memory, the instructions being executable by the processor to:

US 11,140,426 B2

13

14

receive a request for a one-to-many call from a caller
device to a plurality of callee devices;
create a caller port by the media multiplexer to receive
media from the caller device;
create a receiving port to receive media from the 5
plurality of callee devices;
allocate a first port by the media multiplexer on a media
proxy for sending media to the caller device;
allocate a corresponding port by the media multiplexer
on the media proxy for each of the plurality of callee 10
devices that directs media to the media multiplexer;
forward the request to each of the plurality of callee
devices using the corresponding port for each callee
device;
answer the call by each of the plurality of callee devices 15
using a port for receiving media on;
allocate a respective port by the media multiplexer on
the media proxy for each callee answer, wherein
each respective port allocated is used to forward
media to the associated callee device; 20
allocate a second port by the media multiplexer on the
media proxy, wherein the second port allocated
directs caller media to the media multiplexer; and
notify the caller device of the second port on the media
proxy to send media to the media multiplexer. 25

* * * * *

Exhibit D



US007886338B2

(12) **United States Patent
Smith**

(10) **Patent No.:** US 7,886,338 B2
(45) **Date of Patent:** Feb. 8, 2011

(54) **SYSTEM AND METHOD FOR CONTROLLING ACCESS TO LOCAL SERVICES WITHOUT LOSING FAILOVER CAPABILITY**

4,334,171 A	6/1982	Parman et al.
4,393,277 A	7/1983	Besen et al.
4,418,333 A	11/1983	Schwarzbach et al.
4,511,887 A	4/1985	Fiore
4,524,288 A	6/1985	Schimmelpennink et al.
4,527,247 A	7/1985	Kaiser et al.
4,636,791 A	1/1987	Burke et al.
4,644,320 A	2/1987	Carr et al.
4,703,306 A	10/1987	Barrit

(75) Inventor: **W. Eric Smith**, Lindon, UT (US)

(73) Assignee: **Control4 Corporation**, Draper, UT (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 740 days.

(21) Appl. No.: **11/869,638**

(Continued)

(22) Filed: **Oct. 9, 2007**

FOREIGN PATENT DOCUMENTS

(65) **Prior Publication Data**

EP 0 513 443 11/1992

US 2008/0091285 A1 Apr. 17, 2008

Related U.S. Application Data

(Continued)

(60) Provisional application No. 60/849,991, filed on Oct. 6, 2006.

OTHER PUBLICATIONS

Depositor of Bill Nuffer, CPP, Apr. 30, 2005, pp. 1-49.

(51) **Int. Cl.**
H04L 9/32 (2006.01)

Primary Examiner—Hosuk Song

(52) **U.S. Cl.** 726/2; 726/3

(74) *Attorney, Agent, or Firm*—Thorpe North & Western LLP

(58) **Field of Classification Search** 726/2-7, 726/11; 713/189, 193-194; 709/217-225, 709/228-229

(57) **ABSTRACT**

See application file for complete search history.

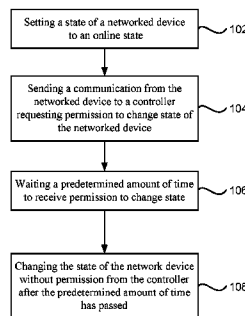
A system and method are provided for controlling access to local electronic devices in an automation network while maintaining failover capability. The method can include the operation of setting a state of a networked device to an online state. Another operation is sending a communication from the networked device to a controller requesting permission to change state of the networked device. The networked device may wait a predetermined amount of time to receive permission to change state. The state of the network device can change without permission from the controller after the predetermined amount of time has passed.

(56) **References Cited**

U.S. PATENT DOCUMENTS

3,878,377 A	4/1975	Brunone
3,909,826 A	9/1975	Schildmeier et al.
4,040,060 A	8/1977	Kaloi
4,068,289 A	1/1978	Ferrigno
4,138,684 A	2/1979	Kerr
4,163,218 A	7/1979	Wu
4,186,381 A	1/1980	Fleischer et al.
4,222,564 A	9/1980	Allen et al.
4,322,842 A	3/1982	Martinez

13 Claims, 3 Drawing Sheets



US 7,886,338 B2

Page 2

U.S. PATENT DOCUMENTS					
			5,224,648 A	7/1993	Simon et al.
			5,237,264 A	8/1993	Moseley et al.
4,706,274 A	11/1987	Baker et al.	5,321,736 A	6/1994	Beasley
4,755,792 A	7/1988	Pezzolo et al.	5,348,078 A	9/1994	Dushane et al.
4,764,981 A	8/1988	Miyahara et al.	5,359,625 A	10/1994	Vander Mey et al.
4,768,218 A	8/1988	Yorita	5,382,947 A	1/1995	Thaler et al.
4,823,069 A	4/1989	Callahan et al.	5,386,209 A	1/1995	Thomas
4,825,209 A	4/1989	Sasaki et al.	5,400,041 A	3/1995	Strickland
4,829,554 A	5/1989	Barnes et al.	5,400,246 A	3/1995	Wilson et al.
4,864,588 A	9/1989	Simpson et al.	5,426,439 A	6/1995	Grossman
4,873,711 A	10/1989	Roberts et al.	5,455,464 A	10/1995	Gosling
4,881,259 A	11/1989	Scordato	5,495,406 A	2/1996	Kushiro et al.
4,889,999 A	12/1989	Rowen	5,530,322 A	6/1996	Ference et al.
4,896,370 A	1/1990	Kasparian et al.	5,544,036 A	8/1996	Brown, Jr. et al.
4,908,604 A	3/1990	Jacob	5,565,855 A	10/1996	Knibbe
4,918,432 A	4/1990	Pauley et al.	5,570,085 A	10/1996	Bertsch
4,918,717 A	4/1990	Bissonnette et al.	5,572,438 A	11/1996	Ehlers et al.
4,928,778 A	5/1990	Tin	5,574,748 A	11/1996	Vander Mey et al.
4,939,792 A	7/1990	Urbish et al.	5,598,039 A	1/1997	Weber
4,940,964 A	7/1990	Dao	5,736,965 A	4/1998	Mosebrook et al.
4,977,529 A	12/1990	Gregg et al.	5,793,300 A	8/1998	Suman et al.
4,996,703 A	2/1991	Gray	5,798,581 A	8/1998	Keagy et al.
5,017,837 A	5/1991	Hanna et al.	5,889,935 A *	3/1999	Ofek et al. 714/6
5,051,720 A	9/1991	Kittirutsunetorn	5,905,442 A	5/1999	Mosebrook et al.
5,079,559 A	1/1992	Umetsu et al.	5,982,103 A	11/1999	Mosebrook et al.
5,086,385 A	2/1992	Launey et al.	6,687,487 B1	2/2004	Mosebrook et al.
5,089,974 A	2/1992	Demeyer et al.	6,803,728 B2	10/2004	Balasubramaniam
5,099,193 A	3/1992	Moseley et al.	2007/0255861 A1 *	11/2007	Kain et al. 710/8
5,109,222 A	4/1992	Welty			
5,146,153 A	9/1992	Luchaco et al.			
5,173,855 A	12/1992	Nielsen et al.			
5,191,265 A	3/1993	D'Aleo et al.			
5,218,344 A	6/1993	Ricketts			
5,218,628 A	6/1993	Ito			

FOREIGN PATENT DOCUMENTS

EP 0 513 688 11/1992

* cited by examiner

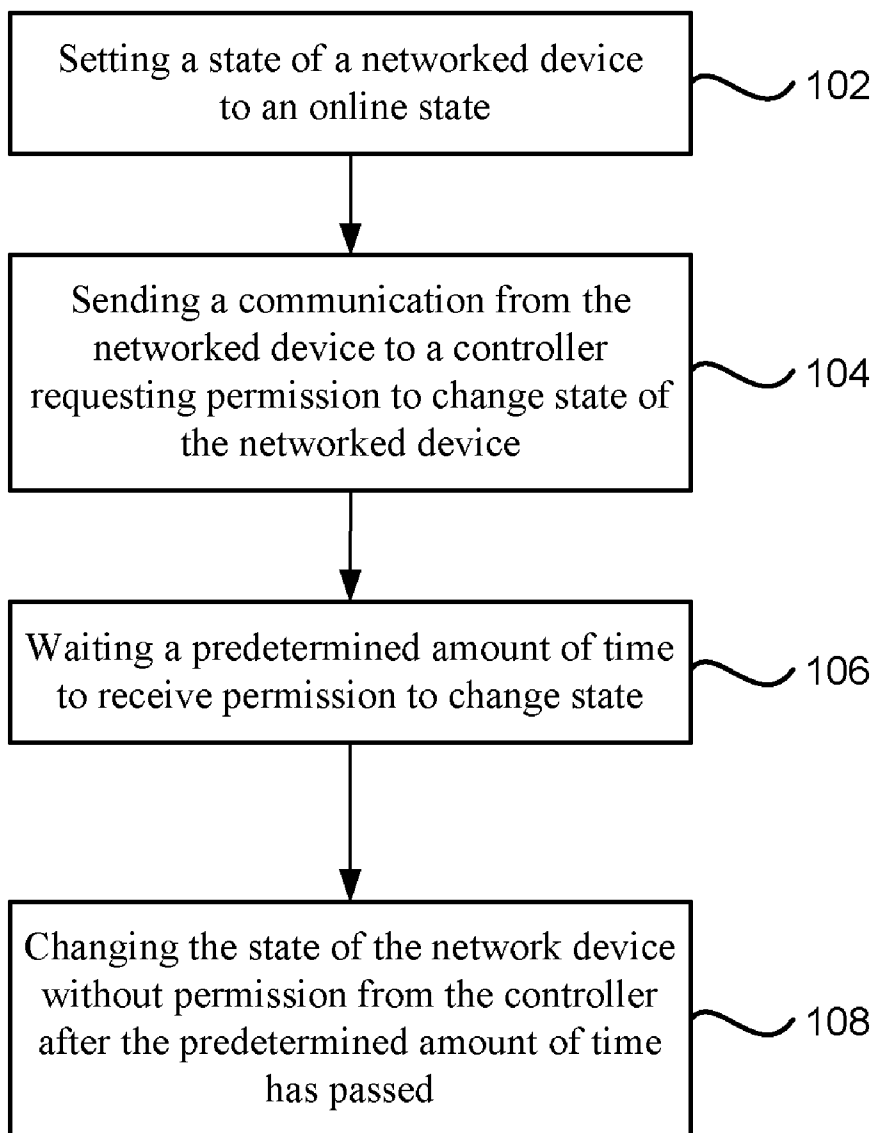


FIG. 1

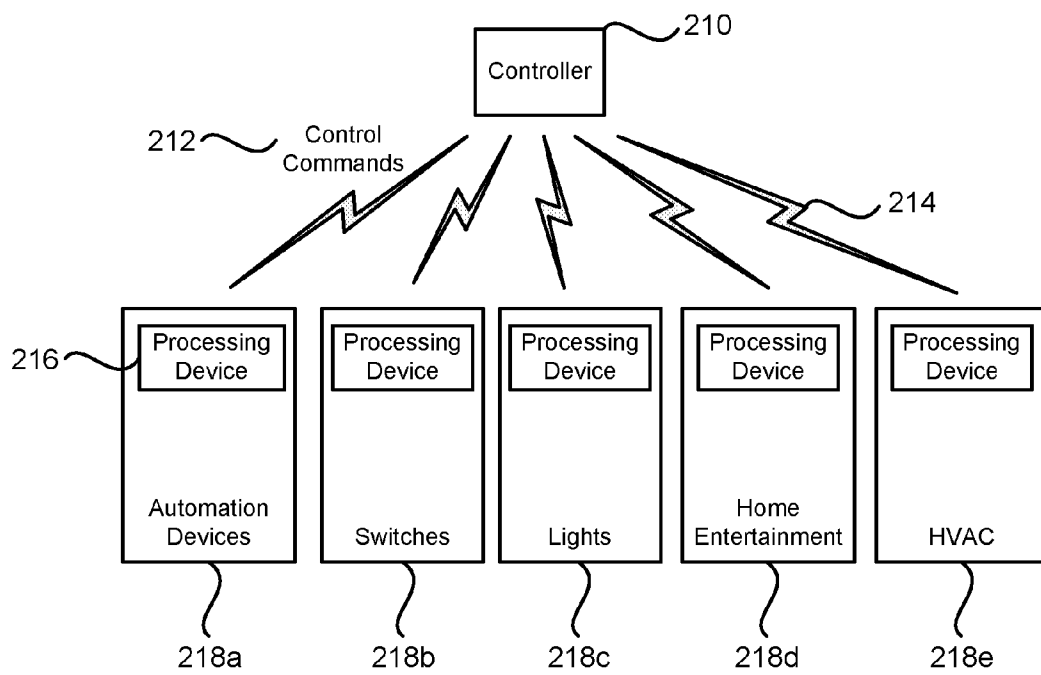


FIG. 2

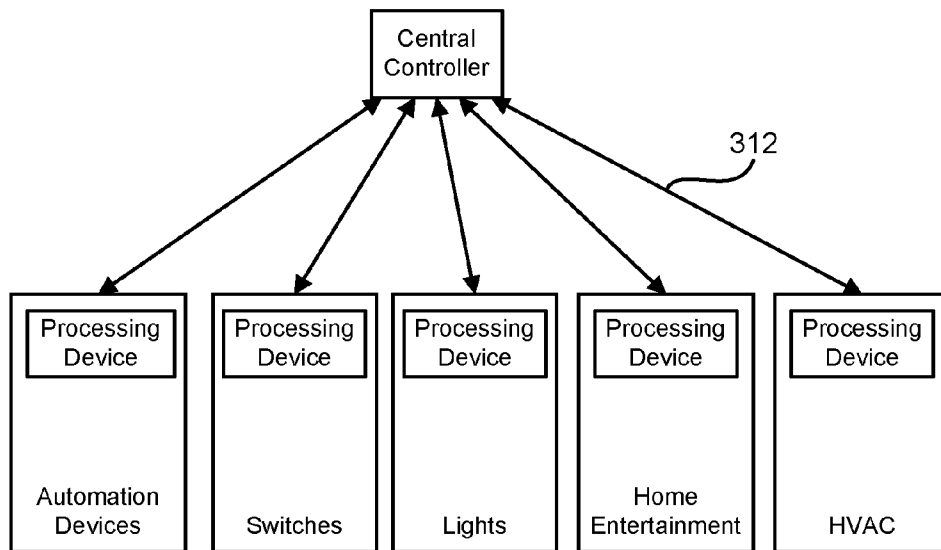


FIG. 3

US 7,886,338 B2

1

**SYSTEM AND METHOD FOR
CONTROLLING ACCESS TO LOCAL
SERVICES WITHOUT LOSING FAILOVER
CAPABILITY**

CLAIM OF PRIORITY

Priority of U.S. Provisional patent application Ser. No. 60/849,991 filed on Oct. 6, 2006 is claimed.

FIELD OF THE INVENTION

The present invention relates generally to control of electronic devices.

BACKGROUND

The price of electronic devices has continued to decrease dramatically. In addition, the variety of consumer electronic components and home automation electronics that can be purchased has continued to increase. Examples of home automation electronics that are available include: remotely controlled light switches and dimmers, wireless networks, wireless power controls, wireless switches, remote controlled fireplaces, garage doors and even appliances. The decrease in prices and increased availability has made home audio, home video, and home automation electronics more available to end users.

The availability of home automation devices, large screen TVs, surround sound stereo equipment, DVDs, CDs, MP3s and similar electronic equipment has also increased consumer interest in networking such devices together. It has become easier and less expensive to buy many networkable devices that can be used to control lighting, monitor appliances, provide high quality movies, music, broadcast television, satellite programs, cable programs, Internet connections, and other electronic components for a home automation network.

Accordingly, there are controller systems and/or switching systems that enable consumer electronics and automatable devices to be connected into a controller. The switching and routing of signals between multiple networked devices can be controlled through that controller. Users who are able to connect all of their components to such a controller or receiver have the benefit of being able to easily switch back and forth between video feeds, control lights and doors, obtain audio streams, control other home automation devices, and provide routing of signals between their electronic components.

Many home automation devices and other systems can be networked into a central controller through a wired or wireless network. Examples of electronic components that an individual may desire to couple to the controller can include: television screens, computer monitors, video projection systems, CD players, DVD players, VCRs, audio equipment, surround sound stereo equipment, video cameras, microphones and other audio-visual and entertainment equipment. In addition, other home and business equipment can be connected to the central controller such as: USB devices, FireWire devices, serial and parallel connections, fiber optic connections, a computer network using an Ethernet or wireless connection, Internet connections. Other types of devices that can be routed into the controller can include home automation equipment such as: lighting control switches, fireplace relays, dimmers, thermostats, HVAC, timers, alarm systems and other types of home automation equipment which can be connected into the controller. While the term home

2

automation is used herein, these electronic components that have been described can be used with the central controller in other settings, including business, education, government, hotels, churches, broadcasting and entertainment facilities.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a flowchart illustrating an embodiment of a method of controlling access to local electronic devices in an automation network while maintaining failover capability;

FIG. 2 is a block diagram illustrating a system for controlling access to local electronic devices in a wireless automation network while maintaining failover capability; and

FIG. 3 is a block diagram illustrating a system for controlling access to local electronic devices in a wired automation network while maintaining failover capability.

DETAILED DESCRIPTION

Reference will now be made to the exemplary embodiments illustrated in the drawings, and specific language will be used herein to describe the same. It will nevertheless be understood that no limitation of the scope of the invention is thereby intended. Alterations and further modifications of the inventive features illustrated herein, and additional applications of the principles of the inventions as illustrated herein, which would occur to one skilled in the relevant art and having possession of this disclosure, are to be considered within the scope of the invention.

There are occasions in an automated building or home where the user wants to restrict access to different services that are provided through automated, networked devices. An example of this type of automated service would be lighting. It is well known that most people fall asleep more easily in a darkened room. Many times parents want to make sure that their kids do not turn on the lights to play or read at night because the children need to sleep.

One approach to solving this problem in an automation system can be to have all of the light switches report their state to the automation controller whenever there is a state change. In this scenario, the controller can just tell the light to turn back off immediately. In most cases this approach would be sufficient, but many times the user wants to ensure that the service is not available at all (e.g. the light doesn't turn on even for a second). This system and method can deliver this kind of control.

For the purpose of providing a clear example, this discussion will refer to the lighting scenario described above, but the control system can easily be extended to additional services like controlling TV time, access to music, control of HVAC (heating and air conditioning) systems, or any automated devices that can be connected into a communication network. Most smart light switches change state (i.e. turn on or off, dim to a level etc.) immediately when the local button is pressed. In one embodiment, the device would only behave this way when in an "offline" state.

Typically, the device would be in an "online" state meaning it is in communications with the controller. In the "online" state, the device can send a communications packet to the controller asking for permission to change state, and the networked device is configured to perform that state change after being given permission by the controller. In this approach, the controller can even track the state of the light without the light switch needing to send state information by simply tracking the requests that were allowed.

One result of the approach described above is that if the device loses communications with the controller, it becomes

impossible to turn on the light. There are many reasons why the controller might not be able to respond to requests. These include controller failure, power outage, interference, etc.

As a result, the present system and method can employ a timeout on requests where if the controller does not respond to a request in a predetermined amount of time, the light can revert to “offline” mode and just make the change without waiting any longer for permission. This does introduce the problem that the controller may track an incorrect state for a short period when there is a system problem, but the controller is able to re-gain the knowledge of the networked device’s state when the device comes back “online” after the next state change request. Re-capturing the state of the device may take only a fraction of a second or several seconds where the malfunction is more serious. This approach also takes away the access restrictions or controls for services when the communication link fails, so this approach is most effectively used in scenarios where the communication rarely fails or where the negative consequences of failure are minimal (like the lighting example). In other words, this system and method can be more effective when the control over access to networked devices is not critical.

FIG. 1 is a flowchart illustrating an embodiment of a method of controlling access to local electronic devices in an automation network while maintaining failover capability. The method can include the operation of setting a state of a networked device to an online state, as in block 102. In other words, when the networked device or automation device is in an online state it is configured to request permission from the controller to change state.

Another operation is sending a communication from the networked device to a controller requesting permission to change state of the networked device, as in block 104. The networked device may wait a predetermined amount of time to receive permission from the controller to change state, as in block 106. The state of the network device can change without permission from the controller after the predetermined amount of time has passed, as in block 108. In other words, if a response has not been received from the controller within a defined time frame then the networked device will timeout and change its state to the desired state.

FIG. 2 illustrates a system for controlling access to local electronic devices in an automation network while maintaining failover capability. The system can include networked devices 218a-e that are set to an online state. The networked devices can be automation devices such as: light switches, power switches, home entertainment devices, heating ventilation and air conditioning (HVAC) controllers, garage door openers, door monitors, DVD players, televisions, personal computers, audio equipment, and other automated devices that can be controlled over a network.

A controller 210 or central processor can be configured to receive communications from a networked device requesting permission to change state of a networked device. For example, a light switch can send a communication to the controller making a request to turn a light from “off” to “on” because its physical switch has been changed by a person. In addition, the controller may be generally defined as an electronic system that may include device automation control systems, receiver functions for audio and video, a central processor and user interface, networking controllers, and other automation and media functions.

A communication network can be provided to transport communications or control commands 212 between the networked devices 218a-e and the controller 210. The communications network can be a wireless network 214 as illustrated in FIG. 2. For example, the wireless network and protocols

may use the Zigbee (IEEE 815.15.4) standard to form ad-hoc networks, also referred to as mesh networks. Mesh networks use a plurality of nodes, with each node being able to communicate to other neighboring nodes that are within range. Thus, each node can act as a repeater. This allows wireless systems employing mesh networks to use inexpensive, lower power transceivers since each node only needs to broadcast as far as a neighboring node, which can then pass the data on to another node until the data reaches a controller or other destination connected to a wired system. Other wireless protocols that can be used are the IEEE 802.11 standard, the Bluetooth standard, or other networking communications protocols but these protocols may be less effective in some situations.

Alternatively, the communications network can be a physically wired network 312 as illustrated in FIG. 3. For example, the wired network can be connected together using Ethernet and twisted pair or coaxial cable wiring. Another example may be use of fiber optic network connections. Of course, other physical wiring connection schemes and their associated protocols can be used.

Referring again to FIG. 2, a processing device 216 can be contained in the networked device and the processing device can be configured to wait a predetermined amount of time to receive permission from the controller to change state. If the network device does not receive a response from the central controller after the predetermined amount of time has passed, then the network device can change state without permission from the controller. The pre-determined amount of time can be set by measuring the maximum amount of time that it is expected to take the controller to respond. That maximum expected response time may be set as the threshold amount of time or the threshold can be set as a multiple of the maximum amount of expected time for response.

The processing device can be an integrated circuit (IC) chip that has a processor and a clock on the chip to time how long it takes for messages to get back to the networked device from the controller. Alternatively, the processing device can be just a timing module. Other circuits that use discrete electronic components may also be used to measure the time it takes for a message to come back.

It is to be understood that the above-referenced arrangements are only illustrative of the application for the principles of the present invention. Numerous modifications and alternative arrangements can be devised without departing from the spirit and scope of the present invention. While the present invention has been shown in the drawings and fully described above with particularity and detail in connection with what is presently deemed to be the most practical and preferred embodiment(s) of the invention, it will be apparent to those of ordinary skill in the art that numerous modifications can be made without departing from the principles and concepts of the invention as set forth herein.

The invention claimed is:

1. A method of controlling access to electronic devices in an automation network while maintaining failover capability, comprising:

- setting a state of a networked device to an online state;
- configuring the networked device to request permission from a controller to change state;
- sending a communication from the networked device to the controller requesting permission to change state of the networked device;
- waiting a predetermined amount of time to receive permission to change state; and

US 7,886,338 B2

5

changing the state of the networked device without permission from the controller after the predetermined amount of time has passed.

2. The method as in claim 1, wherein the step of changing the state of the networked device further comprises the step of changing the state of the networked device to an offline state. 5

3. The method as in claim 1, wherein the predetermined amount of time is a timeout.

4. The method as in claim 1, further comprising the step of tracking the state of the networked device by tracking requests that were allowed by the controller. 10

5. A system for controlling access to local electronic devices in an automation network while maintaining failover capability, comprising:

- a networked device that is set to an online state; 15
- a controller configured to receive communications from the networked device requesting permission to change state of the networked device;
- a communication network that is configured to transport communications between the networked device and the controller; and 20
- a processing device in the networked device configured to wait a predetermined amount of time to receive permission to change state, wherein the processing device includes a clock to time a waiting period for receiving permission from the controller and the networked device will change state without permission from the controller after the predetermined amount of time has passed. 25

6. The system as in claim 5, wherein the communication network is a wireless communication network. 30

7. The system as in claim 5, wherein the communication network is a wired communication network.

8. The system as in claim 5, wherein the controller can track the state of the networked device by tracking requests that were allowed by the controller. 35

9. A method of controlling access to a networked device in an automation network while maintaining failover capability, comprising:

- setting a state of a networked device to an online state;

6

- configuring the networked device to request permission from a controller to change state;
- sending a communication from the networked device to the controller requesting permission to change state of the networked device;
- waiting a predetermined amount of time to receive permission to change state;
- changing the state of the networked device without permission from the controller after the predetermined amount of time has passed and no response has been received from the controller; and
- setting the state of the networked device to an offline state.

10. The method as in claim 9, wherein the predetermined amount of time is a timeout period.

11. The method as in claim 9, further comprising the step of tracking the state of the networked device by tracking requests that were allowed by the controller.

12. A system for controlling access to electronic devices in an automation network while maintaining failover capability, comprising:

- an automation device that is set to an online state;
- a controller configured to receive communications from a networked device requesting permission to change state of the networked device;
- a communication network that is configured to transport communications between the networked device and the controller; and
- a timing module in the networked device configured to wait a predetermined amount of time to receive permission to change state, wherein the timing module includes a clock to time a waiting period for receiving permission from the controller and the networked device will change state without permission from the controller after the predetermined amount of time has passed.

13. The system as in claim 12, wherein the controller can track the state of the networked device by tracking requests that were allowed by the controller.

* * * * *

Exhibit E



(12) **United States Patent**
Lane

(10) **Patent No.:** **US 10,756,984 B2**
(45) **Date of Patent:** **Aug. 25, 2020**

(54) **METHOD AND APPARATUS FOR CREATING AND MANAGING NETWORK DEVICE PORT VLAN CONFIGURATIONS**

(71) Applicant: **Pakedge Device & Software, Inc.**,
Hayward, CA (US)
(72) Inventor: **Nathan Dennison Lane**, La Habra, CA
(US)
(73) Assignee: **WirePath Home Systems, LLC**, Salt
Lake City, UT (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 472 days.

(21) Appl. No.: **14/751,978**

(22) Filed: **Jun. 26, 2015**

(65) **Prior Publication Data**
US 2016/0301577 A1 Oct. 13, 2016

Related U.S. Application Data

(60) Provisional application No. 62/146,602, filed on Apr. 13, 2015.

(51) **Int. Cl.**
G06F 3/048 (2013.01)
H04L 12/24 (2006.01)
H04L 12/46 (2006.01)

(52) **U.S. Cl.**
CPC **H04L 41/22** (2013.01); **H04L 12/4675**
(2013.01); **H04L 12/4679** (2013.01)

(58) **Field of Classification Search**
CPC H04L 41/22; G06F 3/0481-0489
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,452,415 A * 9/1995 Hotka H04L 41/5048
709/224
5,541,586 A * 7/1996 Wise H04N 21/2383
340/12.32

(Continued)

OTHER PUBLICATIONS

Netgear, "GS108T and GS110TP Smart Switch Software Administration Manual", ver. 2.0, published on Nov. 2010 by Netgear Inc. http://www.downloads.netgear.com/files/GS108T_GS110TP_SWA_5Nov10.pdf.*

(Continued)

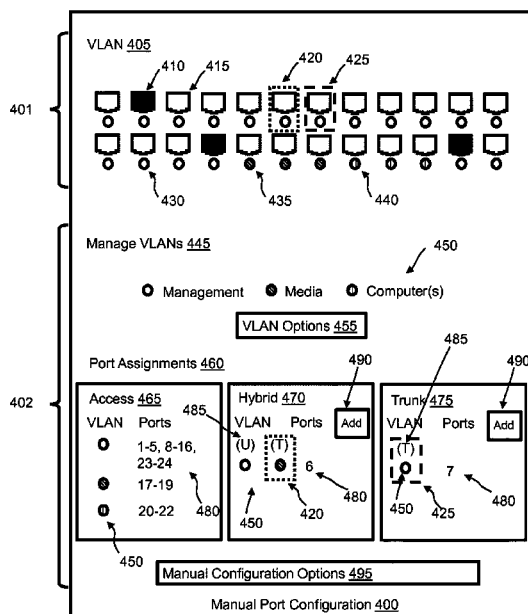
Primary Examiner — Steven B Theriault

(74) *Attorney, Agent, or Firm* — Austin Rapp

(57) **ABSTRACT**

A method for managing VLAN configurations on a network device includes connecting, by a client, to the network device. The method then includes presenting, by the client, a graphical user interface of the network device to a user of the client. The graphical user interface comprises a first area displaying a plurality of ports of the network device and a second area displaying a configuration interface. The configuration interface enables the user to make a VLAN assignment for each of the plurality of ports. Both the first area and the second area are displayed on one page in the graphical user interface. The method then includes accepting, through the graphical user interface, user instructions for making the VLAN assignment. The method then includes transmitting, by the client, a signal to the network device, the signal encoding the user instructions and enabling the network device to make the VLAN assignment.

19 Claims, 5 Drawing Sheets



US 10,756,984 B2

Page 3

(56)	References Cited						
	U.S. PATENT DOCUMENTS						
2008/0304423	A1*	12/2008	Chuang	H04L	12/4666	2014/0280809 A1* 9/2014 Li H04L 41/0806
						370/253	709/222
2008/0309511	A1*	12/2008	Kerr	G06F	1/1613	2014/0281937 A1* 9/2014 Jacobs G06F 3/1208
						340/815.53	715/255
2009/0152216	A1*	6/2009	Champion	G06F	1/183	2016/0044035 A1* 2/2016 Huang H04L 63/0272
						211/26	726/4
2009/0171629	A1*	7/2009	Hirose	H04L	12/4641	2016/0091685 A1* 3/2016 Raza G02B 6/4452
						703/1	398/19
2009/0279549	A1*	11/2009	Ramanathan	G06F	8/65	2016/0134484 A1* 5/2016 Tanaka H04L 41/145
						370/395.4	345/633
2010/0115415	A1*	5/2010	Hickey	H04L	41/0253	2016/0173340 A1* 6/2016 Latham H04L 41/145
						715/736	370/254
2010/0128432	A1*	5/2010	Miller	G06F	1/20	2016/0182293 A1* 6/2016 Di Benedetto H04L 41/0816
						361/679.54	2016/0191314 A1* 6/2016 Russell H04L 41/22
2010/0254374	A1*	10/2010	Fortier	H04Q	1/13	709/221
						370/352	2016/0291838 A1* 10/2016 Tashiro G06F 3/04812
2011/0022695	A1*	1/2011	Dalal	G06F	9/5077	2018/0234306 A1* 8/2018 Pathan H04L 67/10
						709/222	
2011/0058560	A1*	3/2011	Okita	H04L	12/4637	OTHER PUBLICATIONS
						370/395.53	Wikipedia, "Mobile browser", https://en.wikipedia.org/wiki/Mobile_browser .*
2011/0149800	A1*	6/2011	Ryan	H04L	12/467	Wikipedia, "Virtual LAN", https://en.wikipedia.org/wiki/Virtual_LAN .*
						370/254	Cisco, "Using the Web Graphical User Interface", published on Oct. 7, 2013 by Cisco, https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst3650/software/release/3se/ha_stack_manager/configuration_guide/b_hastck_3se_3650_cg/b_hastck_3se_3650_cg_chapter_0101.pdf .*
2011/0209082	A1*	8/2011	Conzola	G06F	17/509	Christhu Raj et al., "A Comprehensive Overview on Different Network Simulators", International Journal of Engineering and Technology vol. 5 No. 1 Feb.-Mar. 2013 (Year: 2013).*
						715/771	Cisco, "UCS Uplink Ethernet Connection Conguration Example", published on Apr. 27, 2010, [online] https://www.cisco.com/c/en/us/support/docs/servers-unified-computing/ucs-6120xp-20-port-fabric-interconnect/110267-ucs-uplink-ethernet-connection.html (Year: 2010).*
2011/0243142	A1*	10/2011	Kwon	H04L	41/5041	Hewlett-Packard Development Company et. al. "HP1910 Gigabit Ethernet Switch Series, User Guide" 2013 (Year: 2013).*
						370/400	Jalan, Raj et. al. "Patch Panel Management using drag and drop and asset templates with v5. 11. 0" Oct. 4, 5, 2015 Retrieved from internet 2019 : https://www.device42.com/blog/2014/10/patch-panel-management-using-drag-and-drop-and-asset-ternplates-with-v5-11-0/ (Year: 2014).*
2011/0302496	A1*	12/2011	Pugaczewski	H04L	41/00	D-Link et. al. "DES-1100-16/24 EasySmart Switch Manual", 2010 (Year: 2010).*
						715/735	D-Link et. al. "DES-1100-10MP Websmart Switch Manual" v 1.0, 2010 (Year: 2010).*
2012/0297379	A1*	11/2012	Anderson	G06F	9/45558	
						718/1	
2013/0058255	A1	3/2013	Casado et al.				
2013/0091534	A1*	4/2013	Gilde	H04L	63/0227	
						726/1	
2013/0117766	A1*	5/2013	Bax	G06F	9/4405	
						719/323	
2013/0223684	A1*	8/2013	Townend	G06Q	10/08	
						382/103	
2013/0227097	A1*	8/2013	Yasuda	H04L	41/0813	
						709/222	
2013/0246673	A1*	9/2013	Cox	H04L	41/0806	
						710/104	
2013/0265882	A1*	10/2013	Li	H04L	43/0805	
						370/241	
2014/0059440	A1*	2/2014	Sasaki	H04L	41/145	
						715/735	
2014/0075320	A1*	3/2014	Sasaki	H04L	12/40	
						715/736	
2014/0137023	A1*	5/2014	Lin	G06F	3/04845	
						715/771	

* cited by examiner

FIG. 1

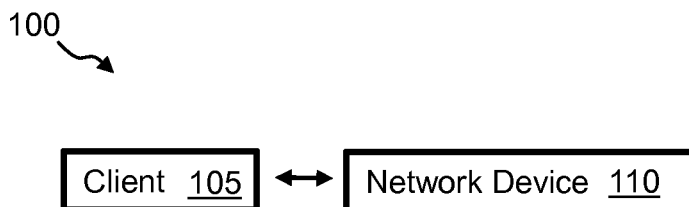


FIG. 2

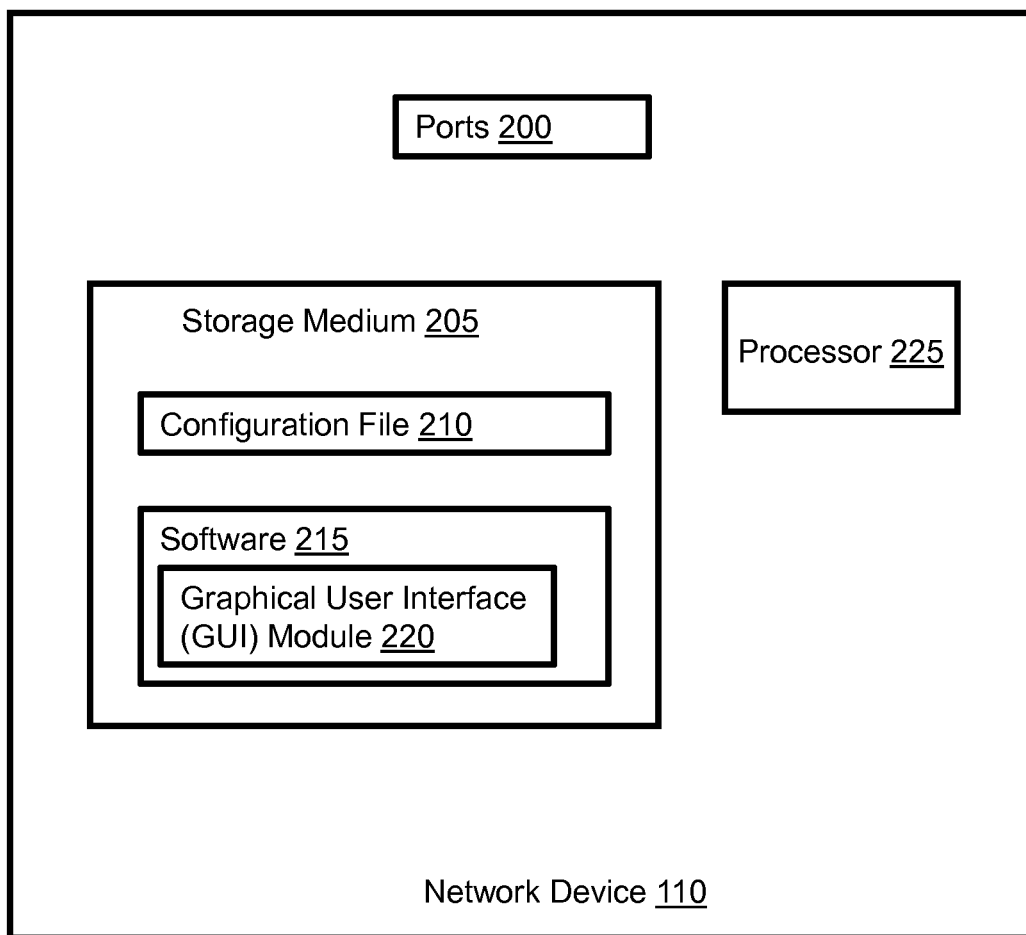


FIG. 3

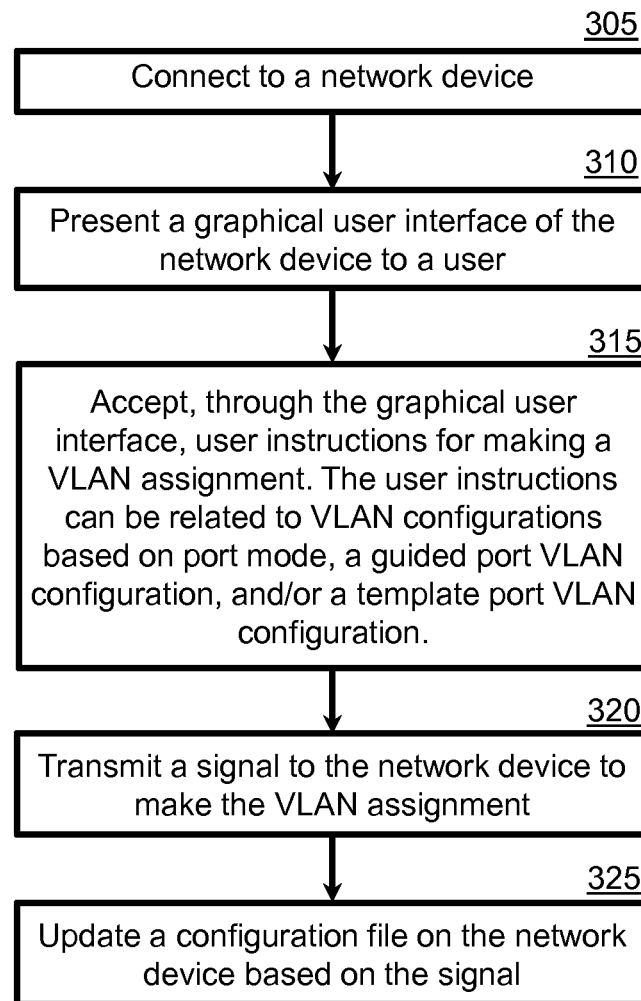


FIG. 4

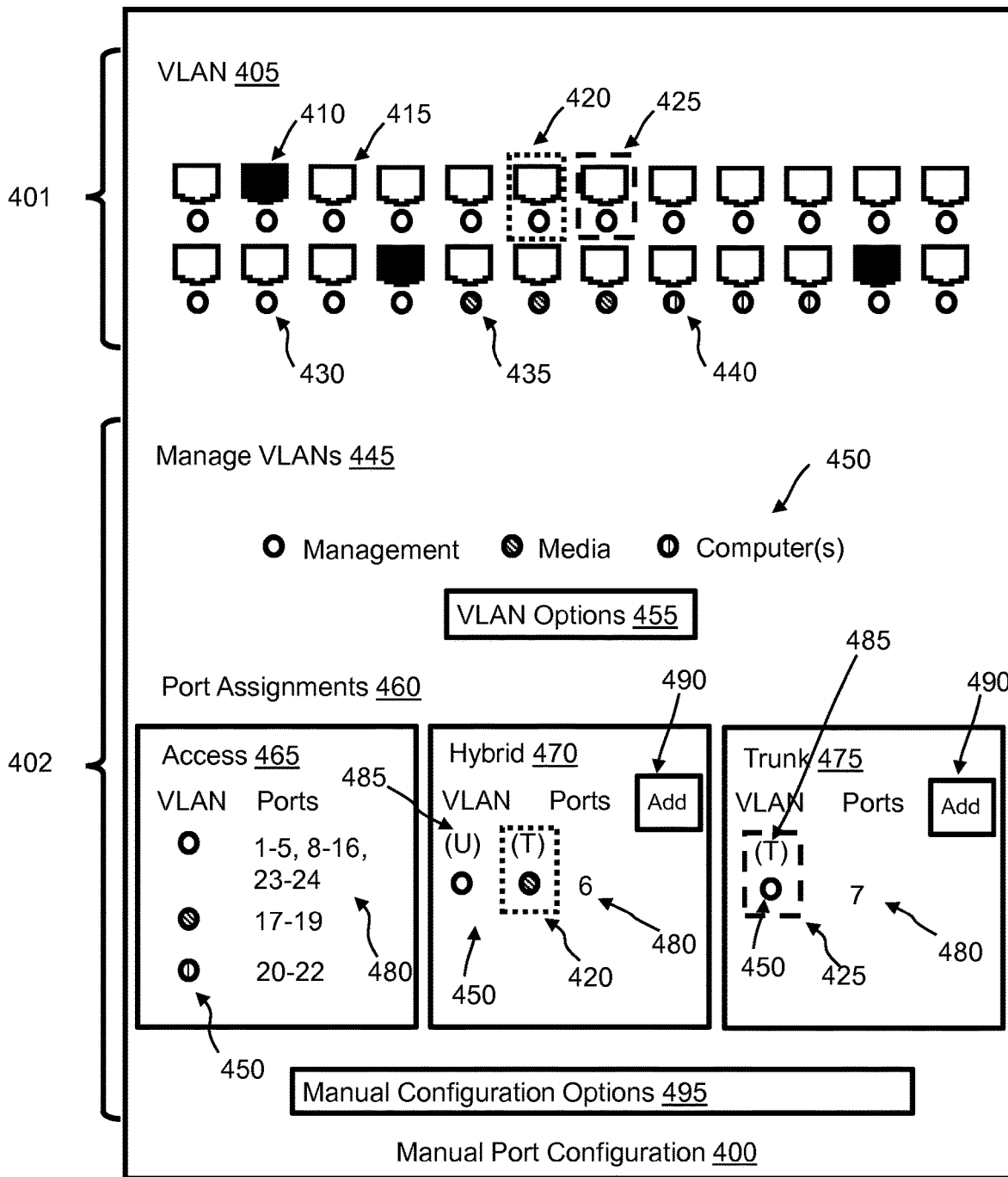


FIG. 5

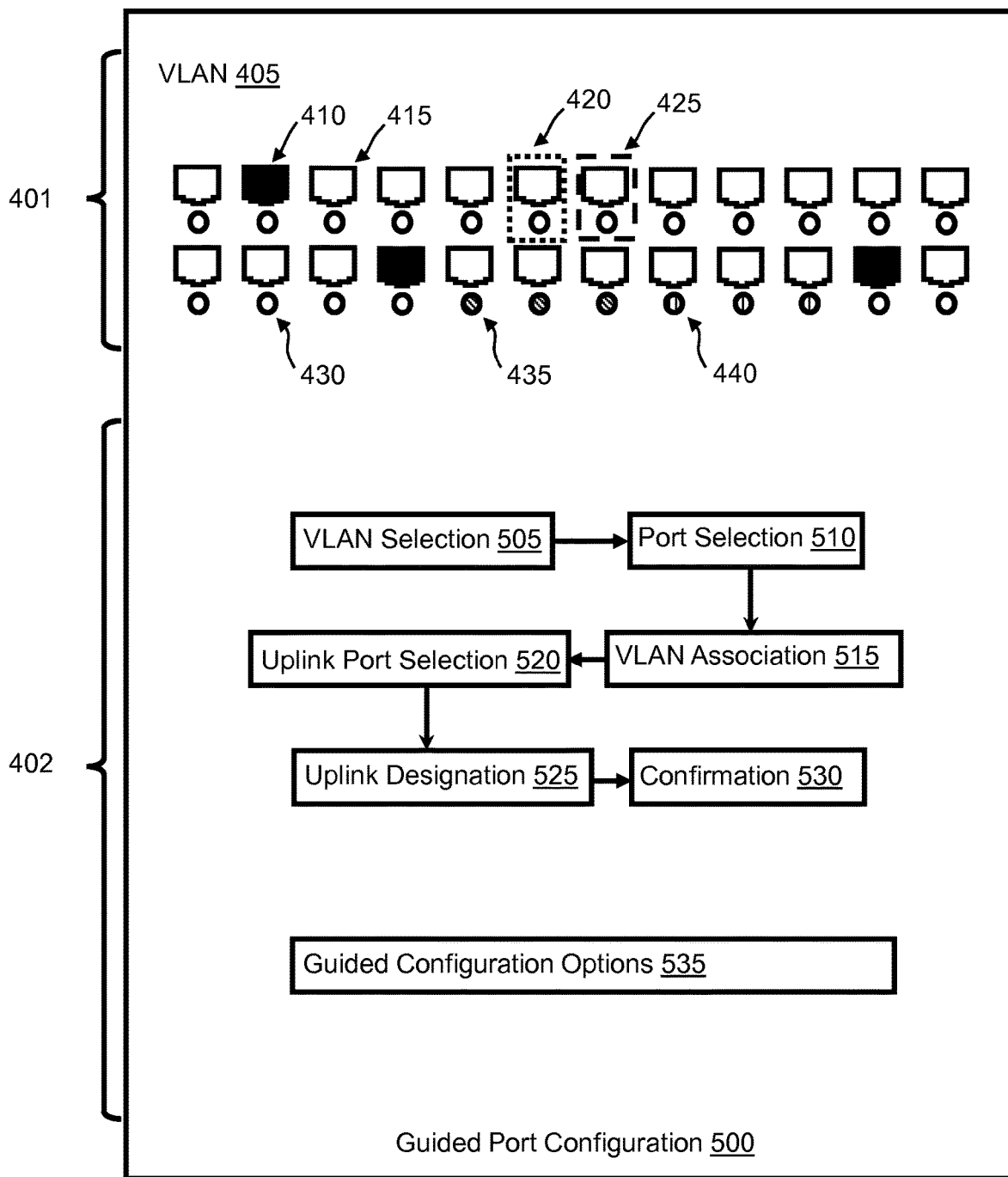
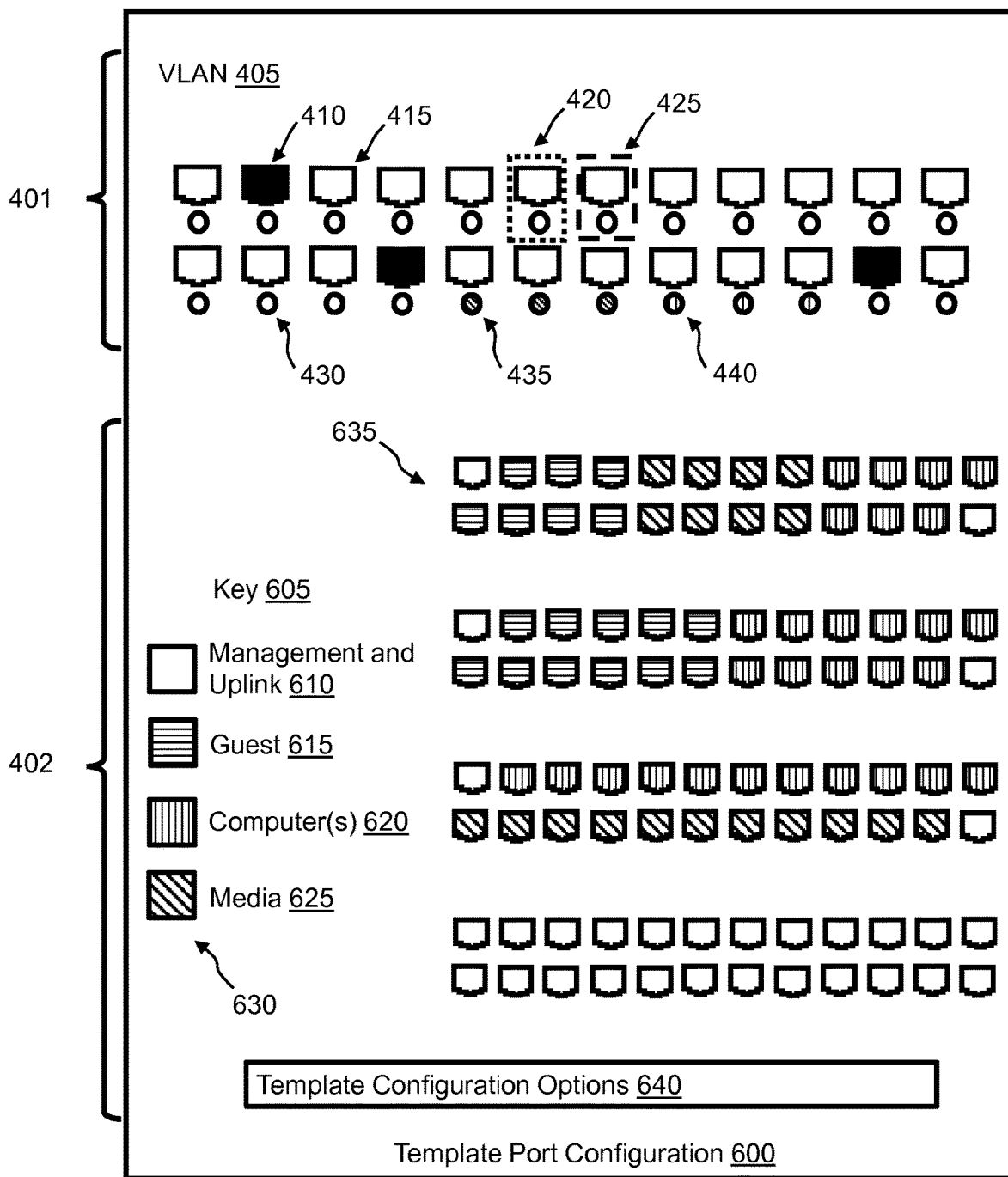


FIG. 6



US 10,756,984 B2

1

METHOD AND APPARATUS FOR CREATING AND MANAGING NETWORK DEVICE PORT VLAN CONFIGURATIONS

CROSS-REFERENCE TO RELATED PATENT APPLICATIONS

This application claims priority from Provisional U.S. Application 62/146,602, filed Apr. 13, 2015, titled “METHOD FOR CREATING AND MANAGING PORT VLAN CONFIGURATIONS,” which is incorporated herein by reference in its entirety.

BRIEF DESCRIPTION OF THE DRAWINGS

Exemplary embodiments are described herein with reference to the following Figures, wherein:

FIG. 1 is a chart showing components of an exemplary system, and exemplary communication between the components.

FIG. 2 shows exemplary components of an exemplary network device.

FIG. 3 is a chart describing an exemplary method.

FIG. 4 shows an exemplary manual port configuration GUI layout.

FIG. 5 shows an exemplary guided port configuration GUI layout and an exemplary guided configuration flow.

FIG. 6 shows an exemplary template port configuration GUI layout.

DETAILED DESCRIPTION

The detailed description set forth below in connection with the appended drawings is intended as a description of various aspects of certain exemplary embodiments and is not intended to represent the only aspects of those embodiments. Each aspect described in this disclosure is provided merely as an example or illustration, and should not necessarily be construed as preferred or advantageous over other aspects. The detailed description includes specific details for providing a thorough understanding of the present disclosure. However, it will be apparent to those skilled in the art that the present disclosure can be practiced without these specific details. Acronyms and other descriptive terminology can be used merely for convenience and/or clarity and are not intended to limit the scope of the present disclosure. Any steps in a method should not necessarily be construed as needing to be carried out in the order listed, unless so stated. The term “can,” used throughout the detailed description, refers generally to something that is possible or is permissible, and should not be viewed as limiting the disclosure to the thing described.

Embodiments of the present disclosure relate generally to configuration of network device ports. Particular embodiments relate to a user interface for creating and managing network device port VLAN configurations. This is achieved using an interactive user interface along with a guided design to streamline complex configuration changes. Although the present disclosure is in reference to network device port VLAN configuration, other port configurations can be performed as well, such as port mode configuration.

According to certain embodiments, a network device can be managed through a client. In some embodiments, a server can intermediate the communication. Many embodiments provide a method comprising communication between the client and the network device, and updating of a configuration

2

file on the network device to reflect user inputs made in the graphical user interface (GUI) of the client.

Previous VLAN configuration methods used a strictly text based approach. The ports had to be selected one at a time for configuration. Information and settings were located across multiple pages instead of on one central VLAN configuration page. The embodiments of the present disclosure, when compared to past methods, provide a faster and easier way to create and modify VLAN configurations on a network device because all of the relevant information, including VLAN visualization and configuration options, are brought to one page. In the present disclosure, “page” refers to a broad area of configuration in a GUI that, when interacted with, does not appear to reload the GUI or to load a new GUI. In embodiments of the present disclosure, the display of configuration settings is also easier to understand. One component on the page provides an easy to understand view of the network device ports, along with their current configuration. Another component on the page shows the multiple VLAN configuration settings. A third component on the page allows the user to configure and change the settings on the network device. The visualization components connect with the configuration components to show current settings and make editing the network device configurations possible.

A graphical user interface of the present disclosure enables port visualization, VLAN setting visualization, and VLAN configuration manipulation (comprising options to add, edit, and/or delete VLANs). A user, who as a non-limiting example can be a network administrator, can create and edit VLANs, can assign VLANs to groups of ports, can configure VLANs as tagged or untagged, and/or can specify access, hybrid, and/or trunk modes for multiple ports all from one page. Applying the access mode to a port configures it for untagged traffic only, with a port VLAN ID set as selected. Applying the hybrid mode to a port configures it to have at least one untagged port VLAN ID, and at least one tagged VLAN ID. Applying the trunk mode to a port configures it for tagged traffic only, with one or more tagged VLAN IDs. All designations and assignments made by the user can then be reflected in a view depicting the ports of the networked device along with their related configurations.

The network device port VLAN configurations can use color coding to match a VLAN with a port. A user can make a variety of inputs in the graphical user interface, including specifying VLAN ID numbers, VLAN name text, and mouse selection input. Some of the outputs that can be displayed by the graphical user interface include: network device port VLAN settings, VLANs added to the system, and VLAN configurations.

When the user connects to the network device and is served up the HTML GUI in their client’s web browser, the user can choose an option to configure VLANs. The user can then select whether they want to do a basic configuration or an advanced configuration. A basic configuration can include a guided port configuration or a template port configuration. An advanced configuration can include a manual port configuration. Each of the configuration options—manual port configuration, guided port configuration, and template port configuration—are disclosed herein. The user can switch between manual, guided, and template configuration options by selecting their desired option from the same page where they perform the configurations. This enables the user to easily select and switch between configuration methods while still remaining on the same page.

Referring to FIG. 1, a chart showing components of system 100, and exemplary communication between the

US 10,756,984 B2

3

components, is shown. System **100** can include a client **105** and a network device **110**. In other embodiments, other components can intermediate the communication between the client **105** and the network device **110**. As a non-limiting example, an intermediary device can be a server. The client **105** can be any device that has networking/web capabilities and a screen to display a graphical user interface. A client **105** can be, as non-limiting examples, a computer or a smartphone. The network device **110** can be any device with ports and with a CPU or an embedded “system on a chip.” In the present embodiment, the network device **110** is a 24 port managed Ethernet switch, with the switch’s HTML GUI being accessed through a web browser on client **105**. However, the present disclosure is not limited to a 24 port managed Ethernet switch.

FIG. 2 shows exemplary components of an exemplary network device **110**. A network device **110** can have, as a non-limiting example, ports **200**, a storage medium **205**, a configuration file **210** and software **215** stored in storage medium **205**, and a processor **225**. Software **215** can have a GUI module **220** which can be used to serve up a graphical user interface to the client **105**. Processor **225** can be a “system on a chip” or a CPU, and can be used to run software **215** to enable a user of client **105** to interact with network device **110**.

Referring to FIG. 3, a chart describing an exemplary method is shown. The method can comprise a step **305** of connecting, by a client **105**, to a network device **110**. Step **310** can comprise presenting a graphical user interface, served up by GUI module **220**, of the network device **110** to a user of the client **105**. Step **315** can comprise accepting, through the graphical user interface, user instructions for making a VLAN assignment. As non-limiting examples, the VLAN assignment can be an access mode port VLAN configuration, a hybrid mode port VLAN configuration, a trunk mode port VLAN configuration, a guided port VLAN configuration, and/or a template port VLAN configuration. Step **320** can comprise transmitting a signal to the network device **110** to make the VLAN assignment. Step **325** can comprise updating configuration file **210** on network device **110** based on the transmitted signal. In such a manner, the method results in a user being able to create and manage VLAN configurations of network device **110**. All of the VLAN assignments can be initiated from one page. This greatly decreases the time needed to create and modify VLAN settings, and also makes the display of configuration information much simpler to understand.

Referring to FIG. 4, an exemplary manual port configuration GUI layout is shown. In other embodiments, the GUI layout can look different and different graphical representations can be used. Client **105** can display a GUI for manual port configuration **400**. The GUI can have a first area **401** showing graphical representation of ports **200** of network device **110**. The GUI can have a second area **402** showing VLAN configuration setting and options for network device **110**. Second area **402** can change as a user switches between different methods of VLAN configuration, but first area **401** can remain largely the same, displaying graphical representations of ports **200** and their current or temporary assigned settings. First area **401** can have a VLAN section **405** which graphically displays the VLANs associated with ports **200** of network device **110**. Ports **200** can be displayed as active ports **410** or inactive ports **415**. Ports can also have a hybrid mode indicator **420** or a trunk mode indicator **425**. In the present embodiment, a default access mode port has no indicator. However, an indicator can be used if desired. Ports can also have a first VLAN indicator **430**, a second VLAN

4

indicator **435**, and a third VLAN indicator **440**. Although the present embodiment shows only three different VLAN indicators, the number of different VLANs and VLAN indicators can range from 1 to as many as the resources of network device **110** will allow. In many embodiments, the maximum number will be even more than the number of ports **200** of network device **110**.

Second area **402** can have a Manage VLANs section **445** and a Port Assignments section **460**. In the exemplary Manage VLANs section **445**, all of the VLANs of network device **110** can be shown. Such VLANs can be shown with VLAN indicators **450**. Examples of VLAN indicators **450** include first VLAN indicator **430**, second VLAN indicator **435**, and third VLAN indicator **440**. However, VLAN indicators **450** can also show a VLAN ID and/or a VLAN name associated with the VLAN, in addition to a graphical representation of the VLAN that can be used in other areas of the graphical user interface. When displayed in the Manage VLANs section **445**, the VLANs are selectable. They can be selected and used in the port configuration methods described herein.

In a non-limiting example, VLAN Options **455** can comprise an option to create a VLAN, an option to delete a VLAN, and an option to edit a VLAN. Creating a VLAN can comprise inputting a VLAN ID and name, and then confirming the input. Deleting a VLAN can comprise selecting a VLAN, selecting the option to delete the VLAN, and then confirming. Editing a VLAN can comprise selecting an option to edit the VLAN, changing the VLAN name, and then confirming. In other embodiments, other options can be available to the user to manage the VLANs of network device **110**. If the user chooses to confirm an option, for example by clicking a “create” button, client **105** can transmit the settings to network device **110** to update configuration file **210** and store the settings.

In an exemplary Port Assignments section **460**, the user can have separate areas to configure ports. There can be a separate Access mode area **465**, Hybrid mode area **470**, and Trunk mode area **475**. All of the created VLANs displayed in the Manage VLANs section **445** can be shown in the Access mode area **465**, with associated VLAN indicators **450**. Each VLAN in Access mode area **465** can be selectable, which when selected would allow the user to choose ports from first area **401** to assign to the selected VLAN. As a non-limiting example, a user can click on multiple ports in first area **401** or can click and drag to select them, then drag from any of the selected ports down to Access mode area **465** to the VLAN they want to assign the ports to. The same method of port assignment can be performed in Hybrid mode area **470** and Trunk mode area **475** as well.

The selected ports would then be displayed as port indicators **480**, and would be associated with VLAN indicators **450**. Alternatively, ports can automatically be populated and displayed as port indicators **480** as soon as the user selects them in first area **401**. The Hybrid mode area **470** and Trunk mode area **475** can also have a tagging indicator **485**, which displays VLAN indicators **450** in a way that enables the user to see and configure which VLANs are tagged and which are untagged. In the present embodiment, in Hybrid mode area **470**, tagging indicator **485** shows text that represents “tagged” and “untagged,” and then places the VLAN indicators **450** in a table-like format that shows which VLAN is associated with which text. Each row in the table can be selectable and editable to specify at least one VLAN that is untagged and at least one VLAN that is tagged. Additionally, there can be an add option **490**, which can be used to add rows to the table, thereby allowing the user to

US 10,756,984 B2

5

make additional port and VLAN configurations. In the present embodiment, since all VLANs are displayed in Access mode area **465**, the rows cannot be deleted and there can be an option to clear a port assignment instead.

In the present embodiment, the Trunk mode area **475** has a similar layout and configuration options to the Hybrid mode area **470**, except that there is no option to specify “untagged” ports and VLANs. The Hybrid mode area **470** and Trunk mode area **475** can also display the corresponding hybrid mode indicator **420** and/or trunk mode indicator **425** around each “tagged” VLAN indicator in the relevant area. The hybrid mode indicator **420** and trunk mode indicator **425** can also be displayed around the relevant configured port in first area **401**. In such a manner, a user can easily see in first area **401**, through the graphical port representations and associated indicators, which port is assigned to which VLAN, which ports are active and which are inactive, and whether the port are in access mode, hybrid mode, or trunk mode. In other embodiments, all indicators can look different, have different colors or shapes, or be located in different areas. The steps for creating the VLANs and configuring the ports can also be different. However, all of the described features should be presented on one page in the manual port configuration **400** GUI.

In the present embodiment, Manual Configuration Options **495** can comprise options to apply changes, cancel changes, clear a port assignment, and/or delete a row or other input from the Port Assignments section **460**. If the user chooses to apply the changes, for example by clicking an “apply” button, client **105** can transmit the settings to network device **110** to update configuration file **210** and store the settings.

Referring to FIG. **5**, an exemplary guided port configuration GUI layout and an exemplary guided configuration flow are shown. In other embodiments, the GUI layout can look different and different graphical representations can be used. Additionally, the guided configuration flow steps can be different or have a different order. Client **105** can display a GUI for guided port configuration **500**. The GUI can have a first area **401** showing graphical representation of ports **200** of network device **110**. The GUI can have a second area **402** showing VLAN configuration setting and options of network device **110**. Second area **402** can change as a user switches between different methods of VLAN configuration, but first area **401** can remain largely the same, displaying graphical representations of ports **200** and their current or temporary assigned settings. First area **401** can have a VLAN section **405** which graphically displays the VLANs associated with ports **200** of network device **110**. Ports **200** can be displayed as active ports **410** or inactive ports **415**. Displayed ports can also have a hybrid mode indicator **420** or trunk mode indicator **425**. In the present embodiment, a default access mode port has no indicator. However, an indicator can be used if desired. Ports can also have a first VLAN indicator **430**, a second VLAN indicator **435**, and a third VLAN indicator **440**. Although the present embodiment shows only three different VLAN indicators, the number of different VLANs and VLAN indicators can range from 1 to as many as the resources of network device **110** will allow. In many embodiments, the maximum number will be even more than the number of ports **200** of network device **110**.

Second area **402** can display different settings and require different inputs at different times, and can also display guided configuration options **535** related to such settings and inputs. The settings, steps, and options in second area **402** can all be part of a guided port and VLAN configuration. In

6

the present embodiment, the first step in guided port configuration **500** is a VLAN selection step **505**. VLAN selection step **505** can comprise display of a graphical representation of pre-defined VLANs, which include pre-defined VLAN IDs and pre-defined VLAN names. The user can have the option to select as many of the pre-defined VLANs as they desire, and then to proceed with the guided configuration. If the user chooses to proceed, second area **402** will display a different setting, step, or option, while still being on the same page as first area **401**. Instead of proceeding, the user can also choose an option to exit the guided configuration so that they can configure ports and VLANs according to another method (but still remaining on the same page). Such options to proceed or exit can be available after each step in guided port configuration **500**.

When the user chooses to proceed, the graphical representation in second area **402** can change to show the pre-defined VLANs that the user selected in VLAN selection step **505**. The second step is a port selection step **510**. Port selection step **510** can comprise the user, in first area **401**, selecting as many displayed ports as they desire. The third step is a VLAN association step **515**. VLAN association step **515** can comprise the user dragging the selected ports into second area **402** and releasing them over a graphical representation of the particular pre-defined VLAN that they would like to assign the ports to. The user can then repeat these selection and dragging steps to assign as many ports as they like to any of the pre-defined VLANs that they chose in VLAN selection step **505**. Once all ports have been assigned a VLAN, the user can have the options to proceed with guided port configuration **500**, to clear any assignments that have been made, and to move back to a previous step in the guide port configuration **500**.

The fourth step is an uplink port selection step **520**. When the user chooses to proceed, the graphical representation in second area **402** can change to show the default uplink ports for network device **110**. Uplink ports can be ports configured in hybrid mode and set with a VLAN ID of 1, along with any VLAN ID set previously as tagged. In the present embodiment, ports **1** and **24** are automatically designated as uplink ports, but any other ports could be so designated. The automatically designated ports are not changeable by the user, thereby keeping the user from locking themselves out of network device **110**. Uplink port selection step **520** can comprise the user, in first area **401**, selecting as many displayed ports as they desire. In the present embodiment, however, ports **1** and **24** cannot be selected because they are automatically set as uplink ports. The fifth step is an uplink port designation step **525**. Uplink port designation step **525** can comprise the user dragging the selected ports into second area **402** and releasing them over the graphical representation of the default uplink ports, thereby changing or adding to the ports that will be designated as uplink ports. The user can have the options to proceed with guided port configuration **500**, to clear any assignments they have made, and to move back to a previous step in the guide port configuration **500**.

The sixth step is a confirmation step **530**. When the user chooses to proceed, the graphical representation in second area **402** can change to show a summary of the VLAN associations and uplink designations made in the previous steps. Confirmation step **530** can comprise the user reviewing which ports they have assigned to which pre-defined VLAN and which ports they have designated as uplink ports. Guided configuration options **535** can comprise an option to confirm the settings (thereby concluding guided port configuration **500**) and an option to move back to a previous

US 10,756,984 B2

7

step in the guided port configuration **500**. If the user chooses to confirm the settings, for example by clicking an “apply” button, client **105** can transmit the settings to network device **110** to update configuration file **210** and store the settings.

In the present embodiment, hybrid mode indicator **420** and trunk mode indicator **425** are displayed in a way that may not be consistent with a guided port configuration, if for example guided port configuration does not provide the user with an option to specify port mode. This is because the display in the present embodiment is of current settings which have been configured with a different method (e.g. the manual port configuration method). The user has not yet gone through the steps of guided port configuration. Once the user begins the process, hybrid mode indicator **420** and trunk mode indicator **425** can disappear and first area **401** can display graphical representations consistent with selections made with the guided port configuration method.

Referring to FIG. **6**, an exemplary template port configuration GUI layout is shown. In other embodiments, the GUI layout can look different and different graphical representations can be used. Client **105** can display a GUI for template port configuration **600**. The GUI can have a first area **401** showing graphical representations of ports **200** of network device **110**. The GUI can have a second area **402** showing VLAN configuration setting and options of network device **110**. Second area **402** can change as a user switches between different methods of VLAN configuration, but first area **401** can remain largely the same, displaying graphical representations of ports **200** and their current or temporary assigned settings. First area **401** can have a VLAN section **405** which graphically displays the VLANs associated with ports **200** of network device **110**. Ports **200** can be displayed as active ports **410** or inactive ports **415**. Ports can also have a hybrid mode indicator **420** or trunk mode indicator **425**. In the present embodiment, a default access mode port has no indicator. However, an indicator can be used if desired. Ports can also have a first VLAN indicator **430**, a second VLAN indicator **435**, and a third VLAN indicator **440**. Although the present embodiment shows only three different VLAN indicators, the number of different VLANs and VLAN indicators can range from 1 to as many as the resources of network device **110** will allow. In many embodiments, the maximum number will be even more than the number of ports **200** of network device **110**.

Second area **402** can display a key **605**, and exemplary pre-defined VLANs in key **605** comprising management and uplink VLAN **610**, guest VLAN **615**, computer(s) VLAN **620**, and media VLAN **625**. Different and/or additional pre-defined VLANs can also be displayed. Each pre-defined VLAN can have a different key indicator **630**. Key indicators **630** can be, as non-limiting examples, different colors or different shapes for each pre-defined VLAN. Second area **402** can also display template options **635**. In the present embodiment, there are four template options **635** that the user can choose from. In other embodiments, one or more template options **635** can be displayed, corresponding to a variety of pre-defined VLANs and port configurations. The pre-defined port configurations can comprise VLAN assignments and uplink port assignments. In the present embodiment, all template options **635** have ports **1** and **24** assigned as uplink ports. The uplink ports can be set as hybrid ports with a VLAN ID of 1 (corresponding to the management VLAN), but can also be tagged ports set to whichever VLAN is selected. The other ports have different VLANs assigned to them, as shown by the different key indicators **630**. In the present embodiment, hybrid mode indicator **420**, trunk mode indicator **425**, first VLAN indicator **430**, second

8

VLAN indicator **435**, and third VLAN indicator **440** displayed in first area **401** are displayed in a way that may not be consistent with any of the template options **635**. This is because those are current settings which have been configured with a different method (e.g. the manual port configuration method). The user has not yet selected or applied a template. Once the user begins the template port configuration method, or selects a template option **635**, the graphical representations in first area **401** will change to match the appropriate key indicator **630** of the selected template option **635**.

In the present embodiment, template configuration options **640** comprise an option to confirm a template selection, and an option to exit so that the user can configure ports and VLANs according to another method. The user can select one of the template options **635** and then confirm the template selection. When they confirm, the client **105** can send the template configuration to the network device **110** and update the configuration file **210** accordingly. In such a manner, the VLAN and uplink port assignments of the selected template option **635** can be configured for network device **110**.

What is claimed is:

1. A method of managing VLAN configurations on a network device, the method comprising:
 - connecting, by a client, to the network device;
 - presenting, by the client, a graphical user interface of the network device to a user of the client, wherein the graphical user interface comprises:
 - a first area displaying a plurality of ports of the network device; and
 - a second area displaying a configuration interface, the configuration interface enabling the user to make a VLAN assignment for each of the plurality of ports, both the first area and the second area being displayed on one page in the graphical user interface, and wherein the graphical user interface enables displaying a port hybrid mode indicator that encloses a configured port in the first area and a matching port hybrid mode indicator that encloses a VLAN indicator of the configured port in the second area;
 - accepting, through the graphical user interface, user instructions for making the VLAN assignment, wherein at least one of the VLAN assignments is made when the user clicks and drags at least one of the ports from the first area to a VLAN in the configuration interface in the second area; and
 - transmitting, by the client, a signal to the network device, the signal encoding the user instructions and enabling the network device to make the VLAN assignment.
2. The method of claim 1, further comprising:
 - in response to the signal, updating, by the network device, a configuration file stored on a storage medium in the network device.
3. The method of claim 1, wherein:
 - the client is a computer and the graphical user interface is presented in a web browser on the computer.
4. The method of claim 1, wherein:
 - the client is a mobile phone and the graphical user interface is presented in a web browser on the mobile phone.
5. The method of claim 1, wherein:
 - the network device is a managed switch.
6. The method of claim 1, further comprising:
 - receiving a first user selection of a VLAN displayed in the configuration interface of the second area;

US 10,756,984 B2

9

receiving a second user selection of at least one of the plurality of ports displayed in the first area; and receiving a confirmation of the first user selection and the second user selection.

7. The method of claim 6, further comprising: prior to receiving the first user selection, receiving a third user selection of an option to create the VLAN; and responsive to receiving the third user selection, creating the VLAN.

8. The method of claim 7, wherein: creating the VLAN comprises, in response to a user input, assigning a VLAN ID and a VLAN name to the VLAN.

9. The method of claim 1, further comprising: receiving a first user selection of a first option to create a hybrid configuration; receiving a second user selection of at least one VLAN from a group of VLANs to designate as untagged; receiving a third user selection of at least one VLAN from the group of VLANs to designate as tagged; receiving a fourth user selection of at least one of the plurality of ports displayed in the first area; and receiving a confirmation of the first user selection, the second user selection, the third user selection, and the fourth user selection.

10. The method of claim 9, further comprising: prior to receiving the first user selection, receiving a fifth user selection of a second option to create the group of VLANs; and responsive to receiving the fifth user selection, creating the group of VLANs.

11. The method of claim 1, further comprising: receiving a first user selection of a first option to create a trunk configuration; receiving a second user selection of at least one VLAN from a group of VLANs to designate as tagged; receiving a third user selection of at least one of the plurality of ports displayed in the first area; and receiving a confirmation of the first user selection, the second user selection, and the third user selection.

12. The method of claim 11, further comprising: prior to receiving the first user selection, receiving a fourth user selection of a second option to create the group of VLANs; and responsive to receiving the fourth user selection, creating the group of VLANs.

13. The method of claim 1, further comprising: receiving a first user selection of at least one VLAN from a pre-defined group of VLANs; receiving a second user selection of at least one of the plurality of ports displayed in the first area; receiving a first user interaction associating the first user selection with the second user selection; receiving a third user selection of at least one port not previously selected; receiving a second user interaction designating the at least one port not previously selected as an uplink port; and receiving a confirmation of the first user selection, the second user selection, the first user interaction, the third user selection, and the second user interaction.

14. The method of claim 1, further comprising: receiving a first user selection of a template, the template comprising: at least one pre-defined VLAN assigned to a group of the plurality of ports; at least one pre-defined color indicator associated with the at least one predefined VLAN; and pre-defined uplink ports; and receiving a confirmation of the first user selection.

10

15. The method of claim 1, wherein: the first area comprises a color indicator for each of the plurality of ports, the color indicator corresponding to a VLAN that each of the plurality of ports has been assigned to.

16. A method of managing VLAN configurations on a network device, the method comprising: receiving a connection, by a network device, from a client; serving up, to the client, a graphical user interface of the network device, wherein the graphical user interface enables a user of the client to input instructions for making a VLAN assignment, and wherein the graphical user interface comprises: a first area displaying a plurality of ports of the network device; and a second area displaying a configuration interface, the configuration interface enabling the user to make the VLAN assignment for each of the plurality of ports, both the first area and the second area being displayed on one page in the graphical user interface, wherein at least one of the VLAN assignments is made when the user clicks and drags at least one of the ports from the first area to a VLAN in the configuration interface in the second area, and wherein the graphical user interface enables displaying a port hybrid mode indicator that encloses a configured port in the first area and a matching port hybrid mode indicator that encloses a VLAN indicator of the configured port in the second area; and receiving, by the network device, a signal from the client, the signal encoding the instructions and enabling the network device to make the VLAN assignment.

17. The method of claim 16, further comprising: in response to the signal, updating, by the network device, a configuration file stored on a storage medium in the network device.

18. A network device comprising: a plurality of ports; a processor; a storage medium; a configuration file stored in the storage medium; and a graphical user interface module, wherein the graphical user interface module is stored in the storage medium and configured to be executed by the processor, the graphical user interface module comprising: instructions for displaying, in a first area of a client device, a graphical representation of the plurality of ports of the network device; instructions for displaying, in a second area of the client device, a configuration interface, the configuration interface enabling a user of the client device to make a VLAN assignment for each of the plurality of ports, wherein at least one of the VLAN assignments is made when the user clicks and drags at least one of the ports from the first area to a VLAN in the configuration interface in the second area, wherein both the first area and the second area are displayed on one page in a graphical user interface of the client device, and wherein the graphical user interface enables displaying a port hybrid mode indicator that encloses a configured port in the first area and a matching port hybrid mode indicator that encloses a VLAN indicator of the configured port in the second area; and instructions for receiving, from the client device, a signal comprising a user input.

US 10,756,984 B2

11

12

19. The network device of claim 18, wherein:
the graphical user interface module further comprises
instructions for updating the configuration file based on
the received signal, thereby making the VLAN assign-
ment.

5

* * * * *

CIVIL COVER SHEET

The JS 44 civil cover sheet and the information contained herein neither replace nor supplement the filing and service of pleadings or other papers as required by law, except as provided by local rules of court. This form, approved by the Judicial Conference of the United States in September 1974, is required for the use of the Clerk of Court for the purpose of initiating the civil docket sheet. (SEE INSTRUCTIONS ON NEXT PAGE OF THIS FORM.)

I. (a) PLAINTIFFS
Snap One, LLC
(b) County of Residence of First Listed Plaintiff Utah
(c) Attorneys (Firm Name, Address, and Telephone Number)
David L. Mortensen, Tanner B. Camp
Foley & Lardner LLP, 95 S. State Street, #2500
Salt Lake City, UT 84111 (801) 401-8900

DEFENDANTS
Josh.ai and AVA, Inc.
County of Residence of First Listed Defendant
NOTE: IN LAND CONDEMNATION CASES, USE THE LOCATION OF THE TRACT OF LAND INVOLVED.
Attorneys (If Known)

II. BASIS OF JURISDICTION (Place an "X" in One Box Only)
1 U.S. Government Plaintiff
2 U.S. Government Defendant
3 Federal Question (U.S. Government Not a Party)
4 Diversity (Indicate Citizenship of Parties in Item III)

III. CITIZENSHIP OF PRINCIPAL PARTIES (Place an "X" in One Box for Plaintiff and One Box for Defendant)
PTF DEF
Citizen of This State 1 1
Citizen of Another State 2 2
Citizen or Subject of a Foreign Country 3 3
Incorporated or Principal Place of Business In This State 4 4
Incorporated and Principal Place of Business In Another State 5 5
Foreign Nation 6 6

IV. NATURE OF SUIT (Place an "X" in One Box Only) Click here for: Nature of Suit Code Descriptions.

Table with columns: CONTRACT, REAL PROPERTY, TORTS, CIVIL RIGHTS, PRISONER PETITIONS, FORFEITURE/PENALTY, LABOR, IMMIGRATION, BANKRUPTCY, SOCIAL SECURITY, FEDERAL TAX SUITS, OTHER STATUTES. Includes various legal categories like Personal Injury, Property Damage, Labor, etc.

V. ORIGIN (Place an "X" in One Box Only)
1 Original Proceeding
2 Removed from State Court
3 Remanded from Appellate Court
4 Reinstated or Reopened
5 Transferred from Another District (specify)
6 Multidistrict Litigation - Transfer
8 Multidistrict Litigation - Direct File

VI. CAUSE OF ACTION
Cite the U.S. Civil Statute under which you are filing (Do not cite jurisdictional statutes unless diversity):
28 U.S.C. § 1338
Brief description of cause:

VII. REQUESTED IN COMPLAINT:
CHECK IF THIS IS A CLASS ACTION UNDER RULE 23, F.R.Cv.P.
DEMAND \$ Unspecified
CHECK YES only if demanded in complaint:
JURY DEMAND: Yes No

VIII. RELATED CASE(S) IF ANY
(See instructions):
JUDGE
DOCKET NUMBER

DATE August 14, 2023 SIGNATURE OF ATTORNEY OF RECORD /s/ David L. Mortensen

FOR OFFICE USE ONLY
RECEIPT # AMOUNT APPLYING IFP JUDGE MAG. JUDGE

INSTRUCTIONS FOR ATTORNEYS COMPLETING CIVIL COVER SHEET FORM JS 44

Authority For Civil Cover Sheet

The JS 44 civil cover sheet and the information contained herein neither replaces nor supplements the filings and service of pleading or other papers as required by law, except as provided by local rules of court. This form, approved by the Judicial Conference of the United States in September 1974, is required for the use of the Clerk of Court for the purpose of initiating the civil docket sheet. Consequently, a civil cover sheet is submitted to the Clerk of Court for each civil complaint filed. The attorney filing a case should complete the form as follows:

- I.(a) Plaintiffs-Defendants.** Enter names (last, first, middle initial) of plaintiff and defendant. If the plaintiff or defendant is a government agency, use only the full name or standard abbreviations. If the plaintiff or defendant is an official within a government agency, identify first the agency and then the official, giving both name and title.
- (b) County of Residence.** For each civil case filed, except U.S. plaintiff cases, enter the name of the county where the first listed plaintiff resides at the time of filing. In U.S. plaintiff cases, enter the name of the county in which the first listed defendant resides at the time of filing. (NOTE: In land condemnation cases, the county of residence of the "defendant" is the location of the tract of land involved.)
- (c) Attorneys.** Enter the firm name, address, telephone number, and attorney of record. If there are several attorneys, list them on an attachment, noting in this section "(see attachment)".
- II. Jurisdiction.** The basis of jurisdiction is set forth under Rule 8(a), F.R.Cv.P., which requires that jurisdictions be shown in pleadings. Place an "X" in one of the boxes. If there is more than one basis of jurisdiction, precedence is given in the order shown below.
 United States plaintiff. (1) Jurisdiction based on 28 U.S.C. 1345 and 1348. Suits by agencies and officers of the United States are included here. United States defendant. (2) When the plaintiff is suing the United States, its officers or agencies, place an "X" in this box.
 Federal question. (3) This refers to suits under 28 U.S.C. 1331, where jurisdiction arises under the Constitution of the United States, an amendment to the Constitution, an act of Congress or a treaty of the United States. In cases where the U.S. is a party, the U.S. plaintiff or defendant code takes precedence, and box 1 or 2 should be marked.
 Diversity of citizenship. (4) This refers to suits under 28 U.S.C. 1332, where parties are citizens of different states. When Box 4 is checked, the citizenship of the different parties must be checked. (See Section III below; **NOTE: federal question actions take precedence over diversity cases.**)
- III. Residence (citizenship) of Principal Parties.** This section of the JS 44 is to be completed if diversity of citizenship was indicated above. Mark this section for each principal party.
- IV. Nature of Suit.** Place an "X" in the appropriate box. If there are multiple nature of suit codes associated with the case, pick the nature of suit code that is most applicable. Click here for: [Nature of Suit Code Descriptions](#).
- V. Origin.** Place an "X" in one of the seven boxes.
 Original Proceedings. (1) Cases which originate in the United States district courts.
 Removed from State Court. (2) Proceedings initiated in state courts may be removed to the district courts under Title 28 U.S.C., Section 1441.
 Remanded from Appellate Court. (3) Check this box for cases remanded to the district court for further action. Use the date of remand as the filing date.
 Reinstated or Reopened. (4) Check this box for cases reinstated or reopened in the district court. Use the reopening date as the filing date.
 Transferred from Another District. (5) For cases transferred under Title 28 U.S.C. Section 1404(a). Do not use this for within district transfers or multidistrict litigation transfers.
 Multidistrict Litigation – Transfer. (6) Check this box when a multidistrict case is transferred into the district under authority of Title 28 U.S.C. Section 1407.
 Multidistrict Litigation – Direct File. (8) Check this box when a multidistrict case is filed in the same district as the Master MDL docket.
PLEASE NOTE THAT THERE IS NOT AN ORIGIN CODE 7. Origin Code 7 was used for historical records and is no longer relevant due to changes in statute.
- VI. Cause of Action.** Report the civil statute directly related to the cause of action and give a brief description of the cause. **Do not cite jurisdictional statutes unless diversity.** Example: U.S. Civil Statute: 47 USC 553 Brief Description: Unauthorized reception of cable service.
- VII. Requested in Complaint.** Class Action. Place an "X" in this box if you are filing a class action under Rule 23, F.R.Cv.P.
 Demand. In this space enter the actual dollar amount being demanded or indicate other demand, such as a preliminary injunction.
 Jury Demand. Check the appropriate box to indicate whether or not a jury is being demanded.
- VIII. Related Cases.** This section of the JS 44 is used to reference related pending cases, if any. If there are related pending cases, insert the docket numbers and the corresponding judge names for such cases.

Date and Attorney Signature. Date and sign the civil cover sheet.