(12) STANDARD PATENT
(19) AUSTRALIAN PATENT OFFICE

(11) Application No. **AU 2015250230 B2**

(54) Title
**Session-based device configuration**

(51) International Patent Classification(s)
***H04L 29/08*** (2006.01)

(21) Application No: **2015250230** (22) Date of Filing: **2015.04.07**

(87) WIPO No: **WO15/164063**

(30) Priority Data

| (31) Number | (32) Date | (33) Country |
|---|---|---|
| **14/257,502** | **2014.04.21** | **US** |

(43) Publication Date: **2015.10.29**
(44) Accepted Journal Date: **2018.11.08**

(71) Applicant(s)
**Microsoft Technology Licensing, LLC**

(72) Inventor(s)
**Menezes, Pascal F.;Hassan, Amer A.;Leeb, Gunter;Haugen, Todd**

(74) Agent / Attorney
**Davies Collison Cave Pty Ltd, Level 15 1 Nicholson Street, MELBOURNE, VIC, 3000, AU**

(56) Related Art
**US 20060218302 A1**

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 October 2015 (29.10.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/164063 A1

(51) International Patent Classification:
*H04L 29/08* (2006.01)

(21) International Application Number:
PCT/US2015/024594

(22) International Filing Date:
7 April 2015 (07.04.2015)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
14/257,502 21 April 2014 (21.04.2014) US

(71) Applicant: **MICROSOFT TECHNOLOGY LICENSING, LLC** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: **MENEZES, Pascal F.**; c/o Microsoft Technology Licensing, LLC, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). **HASSAN, Amer A.**; c/o Microsoft Technology Licensing, LLC, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). **LEEB, Gunter**; c/o Microsoft Technology Licensing, LLC, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US). **HAUGEN, Todd**; c/o Microsoft Technology Licensing, LLC, LCA - International Patents (8/1172), One Microsoft Way, Redmond, Washington 98052-6399 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ,

*[Continued on next page]*

(54) Title: SESSION-BASED DEVICE CONFIGURATION

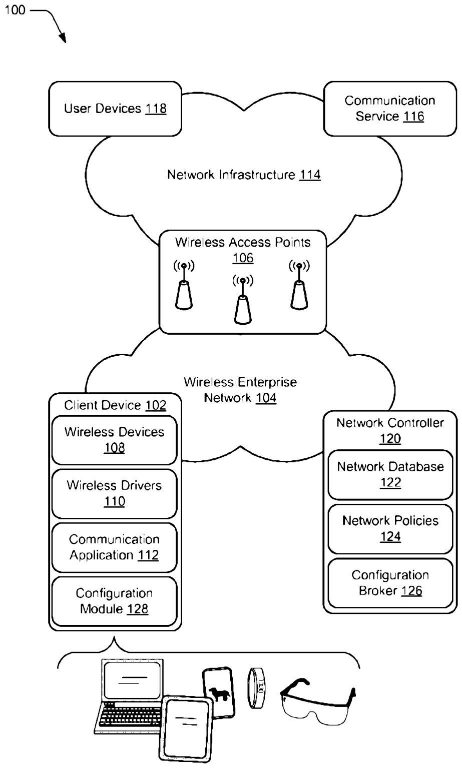

*Fig. 1*

(57) **Abstract**: Techniques for session-based device configuration are described. According to one or more implementations, various settings of a wireless device are configured to optimize device performance while participating in a communication session via a wireless network. The settings, for instance, are configured dynamically and on a per-session basis.

WO 2015/164063 A1

TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

**Declarations under Rule 4.17**:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

**Published**:

— *with international search report (Art. 21(3))*

## BACKGROUND

**[0001]** Mobile computing devices have been developed to increase the functionality that is made available to users in a mobile setting. For example, a user may interact with a mobile phone, tablet computer, or other mobile computing device to check email, surf the web, write texts, interact with applications, and so on. In an enterprise setting, a user may utilize a personal mobile device to engage in enterprise-related activities, such as online meetings, content creation and/or sharing, and so forth.

**[0002]** While allowing a user to utilize their personal device in an enterprise setting is advantageous in terms of cost savings and convenience, it presents a number of implementation challenges. For instance, to leverage an enterprise wireless network to transmit and receive data wirelessly, a personal device typically needs to be configured with particular settings to connect and transmit data over the wireless network. Since a wide variety of different mobile devices exist with a varied assortment of capabilities and operating environments, configuring different devices with the appropriate settings can complicate users' ability to leverage their devices in an enterprise wireless network.

**[0002a]** It is desired to address or ameliorate one or more disadvantages or limitations associated with the prior art, or to at least provide a useful alternative.

## SUMMARY

**[0002b]** According to some embodiments, the present invention provides a system comprising: at least one processor; and one or more computer-readable storage media including instructions stored thereon that, responsive to execution by the at least one processor, cause the system perform operations including: receiving a notification that a communication session is initiated in a network, the notification including a session notification application programming interface (API) that includes a value for an attribute of the communication session; ascertaining that a client device is connected to the network based on the attribute of the communication session received as part of the session notification API; applying the attribute of the communication session received as part of the session notification API to a network policy for the network to specify a parameter for the communication session; configuring a session configuration application programming interface (API) with the parameter for the communication session by applying the value for the attribute included in the session notification API to the network policy; generating a configuration event that includes the session configuration API configured with the parameter for the communication session; and communicating the configuration event to the client device.

**[0002c]** According to some embodiments, the present invention provides a computer-implemented method comprising: receiving a notification that a communication session is initiated in a network, the notification including a session notification application programming interface (API) that includes a value for an attribute of the communication session; ascertaining that a client device is connected to the network based on the attribute of the communication session received as part of the session notification API; applying the attribute of the communication session received as part of the session notification API to a network policy for the network to specify a parameter for the communication session; configuring a session configuration application programming interface (API) with the parameter for the communication session by applying the value for the attribute included in the session notification API to the network policy; generating a configuration event that includes the session configuration API configured with the parameter for the communication session; and communicating the configuration event to the client device.

**[0002d]** According to some embodiments, the present invention provides a method comprising: receiving an update event including an indication of a change to a communication session in a network between a client device and at least one other device, the update event further including a session update application programming interface (API) that includes a value for an attribute of the communication session; ascertaining that the change has occurred in the communication session based on the attribute of the communication session received as part of the session update API; applying the attribute of the communication session received as part of the session update API to a network policy for the network to specify a parameter for the communication session; configuring a reconfiguration application programming interface (API) with the parameter for the communication session by applying the value for the attribute included in the session update API to the network policy; generating a reconfiguration event that includes the reconfiguration API configured with the parameter for the communication session; and communicating the reconfiguration event to the client device.

**[0003]** (Deleted)

**[0004]** (Deleted)

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0005]** Some embodiments of the present invention are hereinafter described, by way of example only, with reference to the accompanying drawings, in which:

**[0006]** FIG. 1 is an illustration of an environment in an example implementation that is operable to employ techniques discussed herein.

**[0007]** FIG. 2 illustrates an example implementation scenario for initiating a communication session in accordance with one or more embodiments.

**[0008]** FIG. 3 illustrates an example implementation scenario for updating session awareness in accordance with one or more embodiments.

**[0009]** FIG. 4 illustrates an example implementation scenario for session termination in accordance with one or more embodiments.

**[0010]** FIG. 5 is a flow diagram that describes steps in a method for applying network policies to a communication session in accordance with one or more embodiments.

**[0011]** FIG. 6 is a flow diagram that describes steps in a method for notifying an entity of communication session attributes in accordance with one or more embodiments.

**[0012]** FIG. 7 is a flow diagram that describes steps in a method for notifying a device of a change in communication session attributes in accordance with one or more embodiments.

**[0013]** FIG. 8 is a flow diagram that describes steps in a method for configuring a device to participate in a communication session in accordance with one or more embodiments.

**[0014]** FIG. 9 illustrates an example system and computing device as described with reference to FIG. 1, which are configured to implement embodiments of techniques described herein.

## DETAILED DESCRIPTION

### Overview

**[0015]** Techniques for session-based device configuration are described. According to one or more implementations, various settings of a wireless device are configured to optimize device performance while participating in a communication session via a wireless network. The settings, for instance, are configured dynamically and on a per-session basis. In at least some embodiments, a communication session refers to an exchange of communication data between different nodes in a network. Examples of a communication session include a Voice over Internet Protocol (VoIP) call, a video call, text messaging, a file transfer, and/or combinations thereof. A communication session, for instance, represents a Unified Communication and Collaboration (UC&C) session.

**[0016]** According to one or more implementations, various settings of a wireless device are configured to optimize device performance while participating in a communication session via an enterprise wireless network. The settings, for instance, are configured dynamically and on a per-session basis.

**[0017]** For instance, consider a scenario where a user device (e.g., a user's personal mobile device) connects to a wireless enterprise network managed by an enterprise entity, 2015250230 26 Sep 2018

such as a business entity, an educational entity, a government entity, and so forth. The enterprise entity establishes various network policies that specify rules and parameters for wireless connections to the enterprise network and/or for participating in communication sessions via the enterprise network.

**[0018]** Further to the example scenario, while connected to the enterprise network, the user's device engages in a communication session with a different device. The different device may be connected to the enterprise network, or may be connected to a different network that communicates with the enterprise network. In response to detecting that the user device is engaging in a communication session, a network controller for the enterprise network ascertains various attributes of the user device and/or the communication session. For instance, the network controller may ascertain the attributes directly from the user device, from network elements of the enterprise network (e.g., wireless access points), and/or via a notification received from an external service.

**[0019]** The network controller applies the attributes to the network policies to specify different configuration parameters for the user device. The configuration parameters, for instance, specify different device settings for the user device. The network controller then generates a notification that includes the configuration parameters. As detailed below, the notification may include an application programming interface (API) that is configured with the parameters.

**[0020]** Further to the example scenario, the network controller communicates the notification to the user device. The user device receives the notification and processes the notification (e.g., the API) to ascertain the configuration parameters. The user device utilizes the configuration parameters to configure various settings and/or attributes of the user device. For instance, the configuration parameters are used to control various wireless-related behaviors, such as off-channel scanning, power saving procedures, wireless access point connections, and so forth.

**[0021]** As referenced above, a device may be configured on a per-session basis, e.g., each time a new communication session is initiated that involves the device. Thus, custom device configurations can be defined (e.g., dynamically and based on network policies) that enable devices to adapt to various network and/or device states, and to dynamically reconfigure themselves based on changes in network policies, network state, device state, and so forth.

**[0022]** In the following discussion, an example environment is first described that is operable to employ techniques described herein. Next, a section entitled "Propagating Session Awareness for Communication Sessions" discusses some example ways for notifying different entities of attributes of communication sessions. Following this, a section entitled "Example Network Policies" describes some example network policies in accordance with one or more embodiments. Next, a section entitled "Example Implementation Scenarios" describes some example implementation scenarios in accordance with one or more embodiments. Following this, a section entitled "Example Procedures" describes some example procedures in accordance with one or more embodiments. Finally, a section entitled "Example System and Device" describes an example system and device that are operable to employ techniques discussed herein in accordance with one or more embodiments.

**[0023]** Having presented an overview of example implementations in accordance with one or more embodiments, consider now an example environment in which example implementations may by employed.

## Example Environment

**[0024]** FIG. 1 is an illustration of an environment 100 in an example implementation that is operable to employ techniques for session-based device configuration described herein. Generally, the environment 100 includes various devices, services, and networks that enable communication via a variety of different modalities. For instance, the environment 100 includes a client device 102 connected to a wireless enterprise network (WEN) 104. The client device 102 may be configured in a variety of ways, such as a traditional computer (e.g., a desktop personal computer, laptop computer, and so on), a mobile station, an entertainment appliance, a smartphone, a netbook, a game console, a handheld device (e.g., a tablet), a wearable computing device, and so forth.

**[0025]** The WEN 104 is representative of a network that provides the client device 102 with connectivity to various networks and/or services, such as the Internet. The WEN 104 may be provided and/or managed by a particular enterprise entity, such as a business entity, an educational institution (e.g., a university), a government institution, and so forth. As used herein, the term "enterprise" generally refers to an entity or group of entities that may maintain a wireless data network for various purposes. The WEN 104 may provide the client device 102 with wireless connectivity via a variety of different connectivity technologies, such as broadband cable, digital subscriber line (DSL), wireless data connectivity (e.g., WiFi™), T-carrier (e.g., T1), Ethernet, and so forth.

**[0026]** The WEN 104 is implemented at least in part via wireless access points (WAP) 106, which are representative of functionality to transmit and receive wireless data as part of the WEN 104. The WAP 106, for instance, provide wireless connectivity for the client device 102 and other wireless-enabled devices. The client device 102 further includes wireless devices 108, which are representative of functionalities to enable the client device

102 to transmit and receive wireless data. Example implementations of the wireless devices 108 include different types of antennas, radios, filters, receivers, transmitters, and so forth.

**[0027]** The wireless devices 108 are generally associated with wireless drivers 110, which are representative of functionality to enable interaction between components of the client device 102 and the wireless devices 108, and vice-versa. For instance, a communication application 112 may leverage the wireless drivers 110 to enable communication data to be transmitted and received via the wireless devices 108.

**[0028]** Generally, the communication application 112 is representative of functionality to enable different forms of communication via the client device 102. Examples of the communication application 112 include a voice communication application (e.g., a VoIP client), a video communication application, a messaging application, a content sharing application, and combinations thereof. The communication application 112, for instance, enables different communication modalities to be combined to provide diverse communication scenarios. According to one or more embodiments, the communication application 112 represents an application that is installed on the client device 102. Additionally or alternatively, the communication application 112 can be implemented as a remote application that is accessible via a web browser, a web application, and so forth.

**[0029]** The environment 100 further includes a network infrastructure 114, which is representative of different connected components that exchange, process, and/or route data among various entities. The network infrastructure 114, for instance, represents different networks and/or sub-networks that can be provided and managed by different entities, such as Internet service providers (ISP). For example, the WAP 106 are connected to the network infrastructure 114 (e.g., by a wired and/or wireless connection) to provide the WAP 106 with network connectivity, such as to the Internet, the web, other enterprise networks, and so forth.

**[0030]** In at least some embodiments, the network infrastructure 114 enables different forms of communication. The network infrastructure 114, for example, enables transmission and receipt of voice data, video data, content data, and so forth. In at least some embodiments, the network infrastructure 114 represents a Unified Communication and Collaboration (UC&C)-enabled network.

**[0031]** Connected to and/or implemented as part of the network infrastructure 114 is a communication service 116, which is representative of a service to perform various tasks for management of communication between the client device 102 and user devices 118. The communication service 116, for instance, can manage initiation, moderation, and termination of communication sessions. Examples of the communication service 116 include a VoIP service, an online conferencing service, a UC&C service, and so forth. In at least some embodiments, the communication service 116 may be implemented as or be connected to a private branch exchange (PBX) in communication with a Public Switched Telephone Network ("PSTN") to enable voice communication between the client device 102 and user devices 118.

**[0032]** According to one or more implementations, the client device 102 is configured to interface with the communication service 116 via the communication application 112 to enable communication between the client device 102 and the user devices 118. The communication application 112, for instance, represents a communication portal that is implemented and managed by the communication service 116 to enable various types of communication.

**[0033]** The environment 100 further includes a network controller 120, which is representative of functionality to manage various aspects of the WEN 104. The network controller 120, for instance, is connected to the WEN 104 and maintains state awareness of different components of the WEN 104. For example, the network controller 120 maintains a mapping of the WAP 106 (e.g., in terms of location) and performance attributes of the WAP 106, such as signal quality for the different WAP 106, quality of service (QoS) attributes of the WAP 106, and so forth. The network controller 120, for instance, may be implemented as a software-defined networking (SDN) controller for managing various aspects of the WEN 104.

**[0034]** According to one or more embodiments, the network controller 120 includes connectivity and logic that accesses routing information for the WEN 104. For instance, the network controller 120 can access an Interior Gateway Protocol (IGP) and/or spanning tree switching topology for the WEN 104. This enables the network controller 120 to identify different data routing paths within the WEN 104, and to map and remap the different routing paths. The network controller 120 stores this information as part of a network database 122, which is representative of functionality to track and store state information for components of the WEN 104.

**[0035]** The network controller 120 may augment the network database 122 with performance data from the WAP 106, such as indications of data flow quality across the individual WAP 106. As further detailed herein, this enables the network controller 120 to make decisions based on quality metrics, and to notify various entities (e.g., the client device

102) of quality metrics for the WAP 106 to enable the entities to make network connectivity decisions.

**[0036]** The network controller 120 further maintains network policies 124, which are representative of different rules and parameters for the WEN 104. The network policies 124, for instance, specify particular behaviors and/or settings for devices that connect to the WEN 104. Examples of different example implementations of the network policies 124 are discussed below.

**[0037]** The network controller 120 is configured to propagate the network policies 124 to different entities via a configuration broker 126. Generally, the configuration broker 126 is representative of functionality to interact with different wireless devices (e.g., the client device 102) to enable the devices to be configured based on the network policies 124. The client device 102, for instance, includes a configuration module 128 which is representative of functionality to interact with the configuration broker 126 and/or other functionalities to enable configuration of the client device 102 for wireless communication via the WEN 104.

**[0038]** For example, the configuration broker 126 can communicate various attributes of the network policies 124 to the configuration module 128. The configuration module 128 can cause the client device 102 to be configured according to the attributes, such as to optimize wireless performance of the client device 102. The configuration module 128 may be implemented in a variety of ways, such as via software, firmware, hardware, and/or combinations thereof. According to one or more implementations, the configuration module 128 can be implemented as a physical layer (PHY) and/or media access control (MAC) layer component of the client device 102. Thus, various techniques discussed herein may be implemented at the PHY and/or MAC layer to configure the client device 102 for a communication session.

**[0039]** The network controller 120 may also enable the WAP 106 to be configured for different communication sessions. For instance, various notifications and operations discussed herein with reference to the client device 102 may also be utilized to notify the WAP 106 of communication session attributes and policies to enable the WAP 106 to be configured for particular communication sessions.

**[0040]** In at least some embodiments, configuration of the client device 102 according to the network policies 124 can occur on a per-session basis, e.g., each time the client device 102 participates in a communication session with another device. Further details concerning configuration of the client device 102 according to different network policies 124 and/or session attributes are discussed below.

**[0041]** According to one or more implementations, the network controller 120 maintains active state awareness of various devices connected to the WEN 104, state conditions of the WEN 104, and of communication sessions that involve the WEN 104. For instance, the network database 122 tracks connectivity attributes of different devices and components within the WEN 104. The network database 122, for example, includes records for active communication sessions and dynamically updates the records, such as based on changes in routing path, changes in connection quality, and so forth. In at least some embodiments, quality metrics from the network database 122 can be used to issue notifications to the client device 102 that enable the client device 102 to adjust to various state changes. Further details and implementations of the various entities of the environment 100 are discussed below.

**[0042]** Having described an example environment in which the techniques described herein may operate, consider now a discussion of example ways of propagating various attributes of communication sessions and network policies in accordance with one or more embodiments.

## Propagating Session Awareness for Communication Sessions

**[0043]** According to various embodiments, techniques can be employed to dynamically enlighten various network components with information about communication sessions. For instance, notification events can be generated that include various attributes of communication sessions. The notification events can be propagated to different entities further to techniques for session-based device configuration discussed herein.

**[0044]** In at least some embodiments, notification events can be configured using a communication application programming interface (API) that can be leveraged to configure and communicate session information to various network components involved in a communication session. For example, the communication API can identify dialogue events and session events which can be populated with respective values for a particular communication session. Consider, for instance, the following events and attributes that may be conveyed via a notification event generated by the communication API:

**[0045]** Dialogue Events – These events apply to various portions of a communication session, such as the start, update, and end of a communication session. A dialogue event can include one or more of the following example attributes.

**[0046]** (1) Timestamp: This attribute can be leveraged to specify timestamps for a start of a communication session, updates that occur during a communication session, and an end (e.g., termination) of a communication session.

**[0047]** (2) Source IP Address: This attribute can be leveraged to specify an IP address for a device that is a source of media during a communication session, e.g., a device that initiates a communication session.

**[0048]** (3) Destination IP Address: This attribute can be leveraged to specify an IP address for a device that is to receive media as part of a communication session.

**[0049]** (4) Transport Type: This attribute can be leveraged to specify a transport type or combination of transport types for a communication session. Examples of transport types include Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and so forth.

**[0050]** (5) Source Port: this attribute can be leveraged to specify an identifier for a port at a source device, e.g., a source device identified by the Source IP Address referenced above.

**[0051]** (6) Destination Port: This attribute can be leveraged to specify an identifier for a port at a destination device, e.g., a destination device identified by the Destination IP Address referenced above.

**[0052]** (7) Media Type: This attribute can be leveraged to specify a media type and/or types that are to be transmitted and/or are being transmitted as part of a communication session. As discussed elsewhere herein, the communication session can involve multiple different types of media. Thus, the Media Type attribute can be employed to identify media types in a communication session, such as for applying network policies discussed herein.

**[0053]** (8) Bandwidth Estimation: This attribute can be leveraged to specify an estimated bandwidth that is to be allocated for a communication session. The estimated bandwidth, for instance, can be based on various factors, such as a privilege level associated with a user, type and/or types of media included in a communication session, a network policy applied to the communication session, and so forth.

**[0054]** (9) To: This attribute can be leveraged to identify a user to which media in a communication session is to be transmitted.

**[0055]** (10) From: This attribute can be leveraged to identify a user from which media in a communication session is transmitted.

**[0056]** (11) Error Code: This attribute can be leveraged to specify various error codes for errors that may occur as part of a communication session. For example, errors can include errors that occur during initiation the communication session, errors that occurred during a communication session, errors that occur when a communication session is terminated, and so forth.

**[0057]** Session Problem Events – These events can be generated and applied when a communication session experiences errors, performance degradation, and so forth. A session problem event may include one or more of the attributes discussed above with reference to Dialogue Events, and may also include one or more of the following attributes.

**[0058]** (1) Mean Opinion Score (MOS) Degradation: This attribute can be leveraged to specify a MOS for a communication session. The attribute, for instance, can be used to indicate that an overall quality of a communication session has decreased.

**[0059]** (2) Jitter Inter-Arrival Time: This attribute can be leveraged to specify jitter values for a communication session. The attribute, for instance, can be used to indicate that a jitter value or values have increased, e.g., have exceeded a specified jitter value threshold.

**[0060]** (3) Packet Loss Rate: This attribute can be leveraged to specify a packet loss rate for a communication session. The attribute, for instance, can be used to indicate that a packet loss rate has increased, e.g., has exceeded a specified packet loss rate value threshold.

**[0061]** (4) Round Trip Delay (RTD): This attribute can be leveraged to specify RTD values for packets in communication sessions. The attribute, for instance, can be used to indicate that RTD values for packets have increased, e.g., have exceeded a specified RTD value threshold.

**[0062]** (5) Concealment Ratio: This attribute can be leveraged to specify a cumulative ratio of concealment time over speech time observed after starting a communication session. The attribute, for instance, can be used to specify that a concealment ratio has increased, e.g., has exceeded a specified concealment ratio value threshold.

**[0063]** Thus, various notifications discussed herein can include one or more of the attributes discussed above and can be used to propagate the attributes to various entities. Elements from the communication API discussed above, for example, can be configured based on network policies and attributes of a communication session. For instance, attributes of a particular communication session can be applied to network policies to configure elements of the communication API. The configured elements can be communicated to a device (e.g., the client device 102) to enable the device to be configured based on values from the communication API elements.

**[0064]** Having described an example ways of propagating session awareness for communication sessions, consider now some example network policies in accordance with one or more embodiments.

## Example Network Policies

**[0065]** The following section describes example network policies (e.g., network policies 124) in accordance with one or more embodiments. As referenced above, network policies generally specify various rules and parameters for connecting to a wireless network, and for transmitting and receiving data via the wireless network.

## Off-Channel Scanning

**[0066]** Generally, off-channel scanning refers to scanning for available wireless network channels. For instance, a device may scan for available wireless channels in attempt to maintain channel awareness in an event that a wireless channel is required.

**[0067]** An example network policy can specify that when a communication session is in progress, off-channel scanning is to be halted and/or minimized. For instance, a network policy may specify that off-channel scanning is not to be performed while a communication session is in progress. Alternatively, a network policy may specify a maximum amount of time during which off-channel scanning may be performed while a communication session is in progress, e.g., 30 milliseconds, 60 milliseconds, and so forth.

**[0068]** In at least some embodiments, a notification event can be sent to a client device notifying the device that the device is currently participating in a communication session, and thus off-channel scanning is to be halted or minimized. The notification event, for instance, can include attributes of the communication API introduced above. When the communication event is terminated, a notification event (e.g., based on the communication API) can be sent to the client device notifying the device that the communication event is terminated, and thus off-channel scanning may resume according to default settings.

## Wireless Mobility

**[0069]** Mobile devices often move between different locations. When a mobile device moves while connected to a wireless network, the mobile device may transfer its network connection between different WAP. For instance, if a user is participating in a communication session with a mobile device while walking between areas of an enterprise facility, handoffs may occur between different WAP to enable the communication session to continue and to maintain an acceptable signal quality.

**[0070]** According to various implementations, network policies can be employed to optimize connection handoff between different WAP. For instance, the network controller 120 can maintain various state information for components of the WEN 104. Examples of such state information include:

**[0071]** (1) An identifier for a current WAP to which the client device 102 is connected.

**[0072]** (2) A location of the client device 102. The location, for instance, can be determined relative to a WAP to which the client device 102 is connected.

**[0073]** (3) Direction of movement of the client device 102. For instance, the network controller 120 can determine that the client device 102 is moving in a particular direction, such as relative to an associated WAP. In at least some embodiments, this information can be received from a WAP that detects movement of the client device 102 in a general direction.

**[0074]** (4) Signal quality attributes of a current connection of the client device 102 to a WAP. Examples of signal quality attributes include signal-to-noise ratio (SNR), received signal strength indicator (RSSI), jitter, packet delay, wireless congestion, and so forth.

**[0075]** (5) Signal quality attributes of other WAP of the WEN 104. The attributes, for instance, can be determined from the WAP themselves, and/or from connected devices.

**[0076]** (6) Locations of other WAP. The network controller 120, for instance, may maintain a map of WAP locations. Further, the map may be augmented with signal quality attributes of the individual WAP such that the network controller 120 maintains a mapping of wireless availability and quality in different locations.

**[0077]** The network controller 120 can utilize this information to enable intelligent decisions to be made regarding access point candidates. For instance, the network controller 120 can identify a best-candidate WAP for the client device 102, e.g., based on location proximity to the client device 102 and signal quality. The network controller 120 can then send a notification event (e.g., using the communication API) to the client device 102 instructing the client device 102 to establish a connection with the WAP.

**[0078]** Alternatively or additionally, the network controller 120 can provide a list of best-candidate WAP to the client device 102, and the client device 102 can employ internal decision-making logic to select a WAP from the list with which to connect.

**[0079]** According to various implementations, this process can occur dynamically and continuously. For instance, the network controller 120 can periodically and/or continuously update its WAP state awareness. Further, the network controller 120 can periodically and/or continuously update the client device 102 regarding best-candidate WAP for wireless data transmission.

## Battery Power and Wireless Performance

**[0080]** Mobile devices often implement battery-saving procedures when operating under battery power. For instance, when disconnected from an alternating current (AC) source, to conserve battery life a mobile device may lower the amount of power used to transmit wireless data. However, reducing the amount of power to a wireless functionality (e.g., the wireless devices 108) may adversely affect wireless signal quality.

**[0081]** Accordingly, a network policy 124 may specify that while a communication session is in progress, power supplied to wireless functionalities is not to be reduced. In at least some implementations, this network policy can override a default device setting that attempts to reduce power for wireless data transmission when a device is operating on battery power.

**[0082]** The network controller 120, for example, can send a notification event to the client device 102 (e.g., using the communication API) indicating that a communication session is in progress, and thus power supplied to wireless functionality is not to be reduced. When the communication session terminates, the network controller 120 can send a notification event to the client device 102 indicating that the communication session has terminated. Thus, the client device may resume default power saving procedures, such as reducing power supplied to wireless functionality.

## Wireless Rate Adaption

**[0083]** Mobile devices may implement rate adaption procedures to compensate for problems in signal quality, such as may occur in areas with noise sources that generate RF interference. Generally, rate adaption refers to a process of reducing a transmission bit rate while increasing transmission power for data transmission. However, typical rate adaption algorithms may adversely affect wireless signal quality. For instance, some rate adaption algorithms cause increases in packet transmission retries and retransmissions, which may cause a receiving device to drop packets as the time sequence to play out media from a communication session expires.

**[0084]** Accordingly, a network policy 124 may specify that while a communication session is in progress, a default rate adaption algorithm is to be overridden with a custom rate adaption algorithm. The custom rate adaption algorithm, for instance, may specify that packet retransmissions and transmission retries are to be reduced from default levels. Implementation of the custom rate adaption algorithm may reduce the likelihood that unnecessary packet retransmissions and transmission retries are performed by a transmitting device.

**[0085]** The network controller 120, for example, can send a notification event to the client device 102 (e.g., using the communication API) indicating that a communication session is in progress, and thus a custom rate adaption algorithm is to be implemented if rate adaption is to be performed. When the communication session terminates, the network controller 120 can send a notification event to the client device 102 indicating that the communication session has terminated. Thus, the client device may resume default rate adaption procedures.

## Quality of Service

**[0086]** According to various implementations, wireless packets that are transmitted may be associated with quality of service (QoS) markings that specify how that packets are to be treated by various network elements. Examples of QoS markings include expedited forwarding, assured forwarding, best effort, and so forth. For instance, a differentiated services code point (DSCP) field in an IP packet can be configured based on different QoS levels to enable different levels of service to be assigned to network traffic. Typical solutions for QoS markings, however, rely on per-packet QoS marking.

**[0087]** Accordingly, a network policy 124 may specify particular QoS levels that are to be applied to transmission of different data packets. The network controller 120, for example, can send a notification event to the client device 102 (e.g., using the communication API) indicating that a communication session is in progress, and thus a particular QoS level is to be applied to packets that are transmitted by the client device 102. The notification event, for instance, is out-of-band from the actual media packets of the communication session. The notification may include actual tags to be applied to the data packets, regardless of how the data packets may be tagged when they are received for transmission. Thus, a QoS level specified by the notification event for packets of a communication session may override a QoS marking attached to the packets. Thus, embodiments discussed herein provide ways of dynamically configuring QoS for communication sessions, such as on a per-session basis.

## Channel Quality

**[0088]** As discussed above, state information regarding different WAP can be maintained, such as location and signal quality for different WAP. Thus, if the client device 102 experiences signal quality degradation with a current WAP, the client device 102 can be informed of candidate replacement WAP. The network controller 120, for example, can send a notification event to the client device 102 (e.g., using the communication API) identifying a candidate WAP and/or wireless channels that the client device 102 may associate with to increase signal quality. In at least some implementations, this can circumvent the need for the client device to perform channel search procedures, such as off-channel scanning.

**[0089]** Having described some example network policies, consider now some example implementation scenarios for session-based device configuration in accordance with one or more embodiments.

## Example Implementation Scenarios

**[0090]** The following section describes example implementation scenarios for session-based device configuration in accordance with one or more embodiments. The implementation scenarios may be implemented in the environment 100 discussed above, and/or any other suitable environment.

**[0091]** FIG. 2 illustrates an example implementation scenario for initiating a communication session generally at 200. The scenario 200 includes various entities and components introduced above with reference to the environment 100.

**[0092]** In the scenario 200, a communication session 202 is initiated between the client device 102 and the user device 118 via the communication service 116. The communication service 116, for instance, serves as an intermediary between the communication application 112 of the client device 102, and the user device 118. For example, the communication service 116 may manage various aspects of initiation, moderation, and termination of the communication session 202.

**[0093]** The communication session 202 may include various types of communication media, such as voice, video, and/or combinations thereof. While the user device 118 is illustrated as being connected outside of the WEN 104, in alternative implementations the client device 102 and the user device 118 may be connected directly to the WEN 104.

**[0094]** In response to initiation of the communication session 202, the communication service 116 generates a notification event 204 and sends the notification event 204 to the network controller 120. The notification event 204 notifies the network controller 120 that the communication session 202 is initiated. The notification event 204 includes a session notification API 206, which represents an implementation of the communication API detailed above.

**[0095]** Further to the scenario 200, the session notification API 206 includes values for various attributes of the communication session 202. Examples of such attributes include identifiers for the client device 102 and the user device 118, such as IP addresses, media access control (MAC) addresses, and so forth. The attributes may further include attributes of the communication session itself, such as a type or types of media being transferred during the communication session, a start time of the communication session, an application ID for the communication application 112, and so forth. Examples of other attributes that may be communicated with the session notification API 206 are detailed above, such as in the discussion of the example communication API and the example network policies.

**[0096]** Thus, based on information from the session notification API (e.g., an ID for the client device 102), the network controller 120 ascertains that the client device 102 is connected to a network domain of the network controller 120. Accordingly, the network controller 120 generates a configuration event 208 that includes a session configuration API 210. The session configuration API 210, for instance, is configured by applying values from the session notification API 206 to the network policies 124.

**[0097]** Further to the scenario 200, the network controller 120 communicates the configuration event 208 to the client device 102 via the WEN 104. For instance, the configuration broker 126 interacts with the configuration module 128 to communicate the configuration event 208. The configuration module 128 includes functionality to consume the session configuration API 210, extract information from the API, and to configure various attributes of the client device 102 based on attributes and values included in the session configuration API 210. For instance, the configuration module 128 can propagate information from the session configuration API 210 to different functionalities of the client device 102 to enable the client device 102 to operate according to the network policies 124, e.g., while engaging in the communication session 202.

**[0098]** As an example, consider that the wireless driver 110 is configured by default to perform periodic off-channel scanning to identify available wireless channels. According to the scenario 200, the session configuration API 210 includes an indication that the client device is either to halt off-channel scanning during the communication session 202, or is to limit the amount of time during which off-channel scanning is performed. The configuration module 128 can read this information from the session configuration API 210, and communicate the information to the wireless driver 110. Thus, the wireless driver 110 may operate according to this policy to limit or stop off-channel scanning while the communication session 202 is active.

**[0099]** This example policy is presented for purpose of example only, and it is to be appreciated that a wide variety of different policies and behaviors can be enforced utilizing techniques discussed herein. Examples of other policies and behaviors that may be utilized are discussed above.

**[00100]** FIG. 3 illustrates an example implementation scenario for updating session awareness, generally at 300. The scenario 300 includes various entities and components introduced above with reference to the environment 100. In at least some embodiments, the scenario 300 represents a continuation of the scenario 200, discussed above.

**[00101]** In the scenario 300, the communication service 116 detects one or more changes in the communication session 202. For instance, the communication service 116 may receive an indication from the client device 102 and/or the user device 118 of a problem with session quality of the communication session 202. Examples of session quality problems include lower than acceptable S/N ratio, low signal strength, too much jitter, too many dropped packets, and so forth.

**[00102]** In response to the indication of session quality problems, the communication service 116 generates an update event 302 that includes a session update API 304. The session update API 304, for instance, represents an implementation of the communication API detailed above. The communication service 116 sends the update event 302 to the network controller 120. The update event 302 notifies the network controller 120 of a change in the communication session 202, e.g., of signal problems with the communication session.

**[00103]** Further to the scenario 300, the session update API 304 includes values for various attributes of the communication session 202. Examples of such attributes include identifiers for the client device 102 and the user device 118, such as IP addresses, media access control (MAC) addresses, and so forth. The attributes may further include a session ID for the communication session and an indication of the change to the communication session. Examples of other attributes that may be communicated with the session update API 304 are detailed above, such as in the discussion of the example communication API and the example network policies.

**[00104]** Thus, based on information from the session update API 304, the network controller 120 ascertains that a problem is occurring with the communication session 202. The session update API 304, for instance, may indicate that signal quality for a WAP 106 to which the client device 102 is connected is poor.

**[00105]** Accordingly, the network controller 120 generates a reconfiguration event 306 that includes a reconfiguration API 308. The reconfiguration API 308, for instance, is configured by applying values from the session update API 304 to the network policies 124. In at least some embodiments, the reconfiguration API 308 may identify candidate WAP 106 that have better signal quality than a current WAP 106 to which the client device 102 is connected.

**[00106]** Further to the scenario 300, the network controller 120 communicates the reconfiguration event 306 to the client device 102 via the WEN 104. For instance, the configuration broker 126 interacts with the configuration module 128 to communicate the reconfiguration event 306. The configuration module 128 includes functionality to consume the reconfiguration API 308, extract information from the API, and to configure various attributes of the client device 102 based on attributes and values included in the reconfiguration API 308. For instance, the configuration module 128 can propagate information from the reconfiguration API 210 to different functionalities of the client device 102 to enable the client device 102 to operate according to the network policies 124, e.g., while engaging in the communication session 202.

**[00107]** In at least some embodiments, based on a candidate WAP 106 identified in the reconfiguration API 308, the client device 102 initiates a handoff procedure to disconnect from a current WAP 106 and to connect to a different WAP 106. Thus, signal quality for the communication session 202 may be increased by connecting to a WAP 106 with higher signal quality.

**[00108]** While the scenario 300 is discussed with reference to the reconfiguration event 306 being generated in response to the update event 302, this is not intended to be limiting. For instance, in at least some embodiments the network controller 120 maintains its own session and/or network awareness independent of the communication service 116. Thus, the network controller 120 can detect changes in network and/or session attributes, and can generate a reconfiguration event and reconfiguration API to notify the client device 102 of the changes and appropriate configuration settings for the client device 102 based on the changes. The network controller 120, for instance, can generate the reconfiguration event 306 and the reconfiguration API 308 based on its own state awareness and independent of a notification from an external entity such as the communication service 116.

**[00109]** Accordingly techniques discussed herein can be employed to dynamically update communication session awareness while a communication session is in progress. Further, update events and reconfiguration events may be issued multiple times during a particular communication session, thus enabling participating devices to be dynamically reconfigured to adapt to changes in session quality and/or session attributes.

**[00110]** FIG. 4 illustrates an example implementation scenario for session termination, generally at 400. The scenario 400 includes various entities and components introduced above with reference to the environment 100. In at least some embodiments, the scenario 400 represents a continuation of the scenarios 200 and 300, discussed above.

**[00111]** In the scenario 400, the communication service 116 detects that the communication session 202 has terminated. For instance, the communication service 116 may receive an indication from the client device 102 and/or the user device 118 that the communication session 202 has ended.

**[00112]** In response to the indication of session termination, the communication service 116 generates an update event 402 that includes a session update API 404. The session update API 404, for instance, represents an implementation of the communication API detailed above. The communication service 116 sends the update event 402 to the network controller 120. The update event 402 notifies the network controller 120 that the communication session 202 has ended.

**[00113]** Further to the scenario 400, the session update API 404 includes values for various attributes of the communication session 202. Examples of such attributes include identifiers for the client device 102 and the user device 118. The attributes may further include a session ID for the communication session 202 and a session end timestamp for the communication session 202. Examples of other attributes that may be communicated with the session update API 404 are detailed above in the discussion of the example communication API.

**[00114]** Thus, based on information from the session update API 404, the network controller 120 ascertains that the communication session 202 has ended. Accordingly, the network controller 120 generates a termination event 406 that includes a termination API 408. The termination API 408, for instance, is configured by applying values from the session update API 404 to the network policies 124. In at least some embodiments, the termination API 408 identifies the communication session 202 and specifies that the communication session has ended.

**[00115]** Further to the scenario 400, the network controller 120 communicates the termination event 406 to the client device 102 via the WEN 104. For instance, the configuration broker 126 interacts with the configuration module 128 to communicate the termination event 406. The configuration module 128 includes functionality to consume the termination API 408 and to configure various attributes of the client device 102 based on attributes and values included in the termination API 408. For instance, the configuration module 128 can propagate information from the termination API 408 to different functionalities of the client device 102 to enable the client device 102 to operate according to the network policies 124.

**[00116]** In at least some embodiments, based on an indication that the communication session 202 is terminated, the client device 102 may notify its various components that they may resume default behavior. For instance, the configuration module 128 may notify the wireless drivers 110 that default behaviors may be resumed, such as with reference to off-channel scanning, battery conservation techniques, wireless rate adaption algorithms, and so forth.

**[00117]** Accordingly techniques discussed herein can be employed to notify devices of session start and stop events, and to dynamically configure device attributes on a per-session basis.

**[00118]** Having discussed some example implementation scenarios, consider now a discussion of some example procedures in accordance with one or more embodiments.

**Example Procedures**

**[00119]** The following discussion describes some example procedures for session-based device configuration in accordance with one or more embodiments. The example procedures may be employed in the environment 100 of FIG. 1, the system 900 of FIG. 9, and/or any other suitable environment. Further, the example procedures may represent implementations of the example scenarios discussed above. In at least some embodiments, steps described for the various procedures can be implemented automatically and independent of user interaction.

**[00120]** FIG. 5 is a flow diagram that describes steps in a method in accordance with one or more embodiments. The method describes an example procedure for applying network policies to a communication session in accordance with one or more embodiments. In at least some implementations, the method can be performed by the network controller 120.

**[00121]** Step 500 receives a notification that a communication session is initiated in a network. The notification, for instance, includes various attributes of the communication session. For example, the notification may be configured via the communication API detailed above. Examples of attributes and information that may be communicated via the notification are described above.

**[00122]** Step 502 ascertains attributes of the communication session from the notification. For example, the network controller 120 can process the notification to identify session attributes, such as from a communication API included with the notification.

**[00123]** Step 504 applies the attributes of the communication session to network policies for the network to specify parameters for the communication session. For instance, different policy-based decisions can be made based on the attributes. Examples of network policies are detailed above.

**[00124]** Step 506 generates a configuration event that includes the parameters for the communication session. The configuration event, for instance, includes a communication API that is populated with various values that represent the parameters for the communication session. Examples of such parameters include behaviors for a device that is participating in the communication session, such as whether to engage in off-channel scanning during the communication session, allowed power conservation techniques during the communication session, QoS marking to be applied to session packets, and so forth.

**[00125]** Step 508 communicates the configuration event to a device that is connected to the network and that is participating in the communication session. In at least some embodiments, information from the configuration event enables the device to configure itself to operate according to the parameters for the communication session.

**[00126]** With reference to the environment 100 and the scenarios discussed above, the network controller 120 can communicate the configuration event to the client device 102. Alternatively or additionally, the network controller 120 can communicate the configuration event to other network elements, such as the WAP 106. For instance, techniques discussed herein may be employed to configure the WAP 106 and/or other network components and network elements, and are not limited to configuration of end-user devices.

**[00127]** FIG. 6 is a flow diagram that describes steps in a method in accordance with one or more embodiments. The method describes an example procedure for notifying an entity of communication session attributes in accordance with one or more embodiments.

**[00128]** Step 600 configures a notification event that includes attributes of a communication session that is occurring in a network. The communication service 116, for instance, populates a communication API with attributes of a communication session. Examples of communication API and communication session attributes are detailed above. In at least some embodiments, the attributes may include attributes of a communication session that was recently initiated, and/or changes to attributes of an existing communication session.

**[00129]** Step 602 communicates the notification event to a network controller for the network. The communication service 116, for instance, communicates the populated communication API to the network controller 120. The notification event may include attributes of a new communication session, and/or changes to attributes of an existing communication session. As detailed herein the network controller 120 can utilize information from the communication API to apply network policies and notify various devices of parameters and behaviors to be applied for a communication session.

**[00130]** FIG. 7 is a flow diagram that describes steps in a method in accordance with one or more embodiments. The method describes an example procedure for notifying a device of a change in communication session attributes in accordance with one or more embodiments.

**[00131]** Step 700 receives an indication of a change in communication session attributes for a communication session that is occurring in a network. The network controller 120, for example, receive an indication that one or more attributes of a communication session have changed. Examples of such a change include a change in session quality, a change in device location, a change in device performance (e.g., for the client device 102 and/or a WAP 106), and so forth. The indication of the change may be received from the communication service 116 and/or based on detected state conditions for the network.

**[00132]** Step 702 generates a reconfiguration event based on the change in the communication session attributes. The network controller 120, for instance, applies the changed attributes to the network policies 124 to generate a session update API for the communication session. The session update API, for instance, includes element values that reflect the change in the communication session attributes as applied to the network policies 124.

**[00133]** In at least some embodiments, the reconfiguration event may identify WAP 106 that are candidates to provide a wireless connection. The candidates may be identified based on signal quality for the individual WAP 106 and/or location for the individual WAP 106. For instance, if the change in the communication session attributes indicates a change in session quality, the reconfiguration event can identify WAP 106 in a particular region that have a higher signal quality than a currently-connected WAP.

**[00134]** Alternatively or additionally, if the change in the communication session attributes indicates that a device (e.g., the client device 102) is moving from one location to another, the reconfiguration event can identify WAP 106 that occur in the general direction of movement and that are available to provide wireless connectivity. Thus, a device that receives the reconfiguration event can process data from the event and select a WAP 106 with which to associate, such as to improve signal quality during a communication event and/or enable the communication event to continue when moving between locations.

**[00135]** Step 704 communicates the reconfiguration event to a device that is connected to the network and that is participating in the communication event. The network controller 120, for instance, communicates the reconfiguration event to the client device 102. Based on information from the communication event, the client device 102 can change its internal settings, can connect to a different WAP 106, and so forth.

**[00136]** FIG. 8 is a flow diagram that describes steps in a method in accordance with one or more embodiments. The method describes an example procedure for configuring a device to participate in a communication session in accordance with one or more embodiments.

**[00137]** Step 800 receives a configuration event that includes parameters to be applied for a communication session. The client device 102, for instance, receives the configuration event from the network controller 120. In at least some embodiments, the configuration event may be an initial configuration event, e.g., a first configuration event that is received after initiation of a communication session. Alternatively, the configuration event may be a reconfiguration event that is received during a communication session and subsequent to a previously-received configuration event for the communication session. According to various implementations, the configuration event is received after the client device 102 has begun participating in the communication session.

**[00138]** Step 802 processes the configuration event to identify the parameters for the communication session. The configuration event, for example, includes a communication API that is populated with different values for different session parameters and/or device settings. The client device 102 may process the communication API to expose the different parameters for the communication session.

**[00139]** Step 804 configures a device for the communication session based on the parameters. The client device 102, for instance, can configure various device settings based on the parameters. For example, the configuration module 128 can communicate various parameters and/or settings to the wireless drivers 110 to enable the wireless drivers 110 to control the wireless devices 108 according to the parameters and settings. Examples of different device settings and attributes that can be configured are discussed above, and include off-channel scan settings, power conservation settings, QoS marking to be applied to communication session packets, and so forth.

**[00140]** A device may be configured as part of an initial configuration of the device for a communication session and/or as part of a configuration update. For instance, the parameters may include updates to previously configured settings and device attributes, such as received as part of a reconfiguration event. Thus, previously-applied settings and attributes for a device participating in a communication session may be updated for the communication session, such as to reflect changes in the communication session.

**[00141]** As referenced above in the discussion of environment 100, the configuration module 128 can be implemented as a PHY and/or MAC layer component of the client device 102. Aspects of the various procedures discussed above, for instance, may be implemented at the PHY and/or MAC layer to configure a device for a communication session. For example, processing of the communication API may occur at the PHY and/or MAC layer to enable various device parameters and settings to be configured for a communication session.

**[00142]** While the method discussed above is described with reference to configuration a user device (e.g., the client device 102) for a communication session, this is not intended to be limiting. For instance, in at least some embodiments, network components such as wireless access points, network firewalls, and so forth, may be configured utilizing techniques discussed herein. Different events and APIs discussed herein, for example, may be communicated to different network components to enable the components to be configured for particular communication sessions. Configuration of network components may occur additionally or alternatively to configuration of an end-user device, and in at least some embodiments may occur in parallel with configuration of an end-user device. For instance, the various notification events discussed above as being communicated to the client device 102 may additionally or alternatively be communicated to one or more of the WAP 106, a network firewall component, a hub, a switch, a router, and so forth, to enable the different components to be configured according to techniques discussed herein.

**[00143]** As discussed above, the different notification events and APIs referenced herein may be communicated separately from data packets of a communication session. Thus, the notification events may be considered as out-of-band communications with regard to communication sessions. In at least some embodiments, this enables devices to be configured and reconfigured for a communication session without interfering with the communication session itself, e.g., independent of a flow of data packets for the communication session.

**[00144]** Having discussed some example procedures, consider now a discussion of an example system and device in accordance with one or more embodiments.

## Example System and Device

**[00145]** FIG. 9 illustrates an example system generally at 900 that includes an example computing device 902 that is representative of one or more computing systems and/or devices that may implement various techniques described herein. For example, the client device 102, the communication service 116, and/or the network controller 120 discussed above can be embodied as the computing device 902. The computing device 902 may be, for example, a server of a service provider, a device associated with the client (e.g., a client device), an on-chip system, and/or any other suitable computing device or computing system.

**[00146]** The example computing device 902 as illustrated includes a processing system 904, one or more computer-readable media 906, and one or more Input/Output (I/O) Interfaces 908 that are communicatively coupled, one to another. Although not shown, the computing device 902 may further include a system bus or other data and command transfer system that couples the various components, one to another. A system bus can include any one or combination of different bus structures, such as a memory bus or memory controller, a peripheral bus, a universal serial bus, and/or a processor or local bus that utilizes any of a variety of bus architectures. A variety of other examples are also contemplated, such as control and data lines.

**[00147]** The processing system 904 is representative of functionality to perform one or more operations using hardware. Accordingly, the processing system 904 is illustrated as including hardware element 910 that may be configured as processors, functional blocks, and so forth. This may include implementation in hardware as an application specific integrated circuit or other logic device formed using one or more semiconductors. The hardware elements 910 are not limited by the materials from which they are formed or the processing mechanisms employed therein. For example, processors may be comprised of semiconductor(s) and/or transistors (e.g., electronic integrated circuits (ICs)). In such a context, processor-executable instructions may be electronically-executable instructions.

**[00148]** The computer-readable media 906 is illustrated as including memory/storage 912. The memory/storage 912 represents memory/storage capacity associated with one or more computer-readable media. The memory/storage 912 may include volatile media (such as random access memory (RAM)) and/or nonvolatile media (such as read only memory (ROM), Flash memory, optical disks, magnetic disks, and so forth). The memory/storage 912 may include fixed media (e.g., RAM, ROM, a fixed hard drive, and so on) as well as removable media (e.g., Flash memory, a removable hard drive, an optical disc, and so forth). The computer-readable media 906 may be configured in a variety of other ways as further described below.

**[00149]** Input/output interface(s) 908 are representative of functionality to allow a user to enter commands and information to computing device 902, and also allow information to be presented to the user and/or other components or devices using various input/output devices. Examples of input devices include a keyboard, a cursor control device (e.g., a mouse), a microphone (e.g., for voice recognition and/or spoken input), a scanner, touch functionality (e.g., capacitive or other sensors that are configured to detect physical touch), a camera (e.g., which may employ visible or non-visible wavelengths such as infrared frequencies to detect movement that does not involve touch as gestures), and so forth. Examples of output devices include a display device (e.g., a monitor or projector), speakers, a printer, a network card, tactile-response device, and so forth. Thus, the computing device 902 may be configured in a variety of ways as further described below to support user interaction.

**[00150]** Various techniques may be described herein in the general context of software, hardware elements, or program modules. Generally, such modules include routines, programs, objects, elements, components, data structures, and so forth that perform particular tasks or implement particular abstract data types. The terms “module,” “functionality,” and “component” as used herein generally represent software, firmware, hardware, or a combination thereof. The features of the techniques described herein are platform-independent, meaning that the techniques may be implemented on a variety of commercial computing platforms having a variety of processors.

**[00151]** An implementation of the described modules and techniques may be stored on or transmitted across some form of computer-readable media. The computer-readable media may include a variety of media that may be accessed by the computing device 902. By way of example, and not limitation, computer-readable media may include “computer-readable storage media” and “computer-readable signal media.”

**[00152]** “Computer-readable storage media” may refer to media and/or devices that enable persistent storage of information in contrast to mere signal transmission, carrier waves, or signals per se. Computer-readable storage media do not include signals per se. The computer-readable storage media includes hardware such as volatile and non-volatile, removable and non-removable media and/or storage devices implemented in a method or technology suitable for storage of information such as computer readable instructions, data structures, program modules, logic elements/circuits, or other data. Examples of computer-readable storage media may include, but are not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, hard disks, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or other storage device, tangible media, or article of manufacture suitable to store the desired information and which may be accessed by a computer.

**[00153]** “Computer-readable signal media” may refer to a signal-bearing medium that is configured to transmit instructions to the hardware of the computing device 902, such as via a network. Signal media typically may embody computer readable instructions, data structures, program modules, or other data in a modulated data signal, such as carrier waves, data signals, or other transport mechanism. Signal media also include any information delivery media. The term “modulated data signal” means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media include wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, radio frequency (RF), infrared, and other wireless media.

**[00154]** As previously described, hardware elements 910 and computer-readable media 906 are representative of instructions, modules, programmable device logic and/or fixed device logic implemented in a hardware form that may be employed in some embodiments to implement at least some aspects of the techniques described herein. Hardware elements may include components of an integrated circuit or on-chip system, an application-specific integrated circuit (ASIC), a field-programmable gate array (FPGA), a complex programmable logic device (CPLD), and other implementations in silicon or other hardware devices. In this context, a hardware element may operate as a processing device that performs program tasks defined by instructions, modules, and/or logic embodied by the hardware element as well as a hardware device utilized to store instructions for execution, e.g., the computer-readable storage media described previously.

**[00155]** Combinations of the foregoing may also be employed to implement various techniques and modules described herein. Accordingly, software, hardware, or program modules and other program modules may be implemented as one or more instructions and/or logic embodied on some form of computer-readable storage media and/or by one or more hardware elements 910. The computing device 902 may be configured to implement particular instructions and/or functions corresponding to the software and/or hardware modules. Accordingly, implementation of modules that are executable by the computing device 902 as software may be achieved at least partially in hardware, e.g., through use of computer-readable storage media and/or hardware elements 910 of the processing system. The instructions and/or functions may be executable/operable by one or more articles of manufacture (for example, one or more computing devices 902 and/or processing systems 904) to implement techniques, modules, and examples described herein.

**[00156]** As further illustrated in FIG. 9, the example system 900 enables ubiquitous environments for a seamless user experience when running applications on a personal computer (PC), a television device, and/or a mobile device. Services and applications run substantially similar in all three environments for a common user experience when transitioning from one device to the next while utilizing an application, playing a video game, watching a video, and so on.

**[00157]** In the example system 900, multiple devices are interconnected through a central computing device. The central computing device may be local to the multiple devices or may be located remotely from the multiple devices. In one embodiment, the central computing device may be a cloud of one or more server computers that are connected to the multiple devices through a network, the Internet, or other data communication link.

**[00158]** In one embodiment, this interconnection architecture enables functionality to be delivered across multiple devices to provide a common and seamless experience to a user of the multiple devices. Each of the multiple devices may have different physical requirements and capabilities, and the central computing device uses a platform to enable the delivery of an experience to the device that is both tailored to the device and yet common to all devices. In one embodiment, a class of target devices is created and experiences are tailored to the generic class of devices. A class of devices may be defined by physical features, types of usage, or other common characteristics of the devices.

**[00159]** In various implementations, the computing device 902 may assume a variety of different configurations, such as for computer 914, mobile 916, and television 918 uses. Each of these configurations includes devices that may have generally different constructs and capabilities, and thus the computing device 902 may be configured according to one or more of the different device classes. For instance, the computing device 902 may be implemented as the computer 914 class of a device that includes a personal computer, desktop computer, a multi-screen computer, laptop computer, netbook, and so on.

**[00160]** The computing device 902 may also be implemented as the mobile 916 class of device that includes mobile devices, such as a mobile phone, portable music player, portable gaming device, a tablet computer, a multi-screen computer, and so on. The computing device 902 may also be implemented as the television 918 class of device that includes devices having or connected to generally larger screens in casual viewing environments. These devices include televisions, set-top boxes, gaming consoles, and so on.

**[00161]** The techniques described herein may be supported by these various configurations of the computing device 902 and are not limited to the specific examples of the techniques described herein. For example, functionalities discussed with reference to the communication service 116, the communication application 112, and/or the network controller 120 may be implemented all or in part through use of a distributed system, such as over a “cloud” 920 via a platform 922 as described below.

**[00162]** The cloud 920 includes and/or is representative of a platform 922 for resources 924. The platform 922 abstracts underlying functionality of hardware (e.g., servers) and software resources of the cloud 920. The resources 924 may include applications and/or data that can be utilized while computer processing is executed on servers that are remote from the computing device 902. Resources 924 can also include services provided over the Internet and/or through a subscriber network, such as a cellular or Wi-Fi network.

**[00163]** The platform 922 may abstract resources and functions to connect the computing device 902 with other computing devices. The platform 922 may also serve to abstract scaling of resources to provide a corresponding level of scale to encountered demand for the resources 924 that are implemented via the platform 922. Accordingly, in an interconnected device embodiment, implementation of functionality described herein may be distributed throughout the system 900. For example, the functionality may be implemented in part on the computing device 902 as well as via the platform 922 that abstracts the functionality of the cloud 920.

**[00164]** Discussed herein are a number of methods that may be implemented to perform techniques discussed herein. Aspects of the methods may be implemented in hardware, firmware, or software, or a combination thereof. The methods are shown as a set of steps that specify operations performed by one or more devices and are not necessarily limited to the orders shown for performing the operations by the respective blocks. Further, an operation shown with respect to a particular method may be combined and/or interchanged with an operation of a different method in accordance with one or more implementations. Aspects of the methods can be implemented via interaction between various entities discussed above with reference to the environment 100.

## Conclusion

**[00165]** Techniques for session-based device configuration are described. Although embodiments are described in language specific to structural features and/or methodological acts, it is to be understood that the embodiments defined in the appended claims are not necessarily limited to the specific features or acts described. Rather, the specific features and acts are disclosed as example forms of implementing the claimed embodiments.

**[00166]** Throughout this specification and the claims which follow, unless the context requires otherwise, the word "comprise", and variations such as "comprises" and "comprising", will be understood to imply the inclusion of a stated integer or step or group of integers or steps but not the exclusion of any other integer or step or group of integers or steps.

**[00167]** The reference in this specification to any prior publication (or information derived from it), or to any matter which is known, is not, and should not be taken as an acknowledgment or admission or any form of suggestion that that prior publication (or information derived from it) or known matter forms part of the common general knowledge in the field of endeavour to which this specification relates.

## THE CLAIMS DEFINING THE INVENTION ARE AS FOLLOWS:

1. A system comprising:

at least one processor; and

one or more computer-readable storage media including instructions stored thereon that, responsive to execution by the at least one processor, cause the system perform operations including:

receiving a notification that a communication session is initiated in a network, the notification including a session notification application programming interface (API) that includes a value for an attribute of the communication session;

ascertaining that a client device is connected to the network based on the attribute of the communication session received as part of the session notification API;

applying the attribute of the communication session received as part of the session notification API to a network policy for the network to specify a parameter for the communication session;

configuring a session configuration application programming interface (API) with the parameter for the communication session by applying the value for the attribute included in the session notification API to the network policy;

generating a configuration event that includes the session configuration API configured with the parameter for the communication session; and

communicating the configuration event to the client device.

2. The system as recited in claim 1, wherein the notification is separate from data packets of the communication session.

3. The system as recited in claim 1 or 2, wherein the attribute includes an identifier for the client device.

4. The system as recited in any one of claims 1 - 3, wherein the attribute includes one or more media types for the communication session.

5. The system as recited in any one of claims 1 - 4, wherein the session notification API further includes multiple attributes of the communication session.

6. The system as recited in any one of claims 1 - 5, wherein the parameter for the communication session specifies a wireless behavior to be applied by the client device.

7. The system as recited in any one of claims 1 - 6, wherein the parameter for the communication session specifies a quality of service marking to be applied by the client device to data packets of the communication session.

8. The system as recited in any one of claims 1 - 3, the operations further including: receiving an indication of a change in the communication session; generating a reconfiguration event that includes at least one of a change to the parameter for the communication session or a change to a different parameter for the communication session; and communicating the reconfiguration event to the client device.

9. The system as recited in claim 8, wherein the indication of the change includes an indication of a problem with session quality for the communication session, and wherein at least one of the change to the parameter for the communication session or the change to the different parameter is specified to increase the session quality for the communication session.

10. The system as recited in claim 8, wherein the indication of the change includes an indication that the client device is moving to a different location, and wherein at least one of the change to the parameter for the communication session or the change to the different parameter identifies a wireless access point that is available to provide wireless connectivity at the different location.

11. The system as recited in claim 8, wherein the indication of the change includes an indication that the communication session is terminated, and wherein at least one of the change to the parameter for the communication session or the change to the different parameter includes an indication that the communication session is terminated.

12. The system as recited in any one of claims 1 - 11, wherein the session notification API is configured to identify dialogue events and session events, and populate the dialogue events and session events with respective values for the communication session.

13. A computer-implemented method comprising:

receiving a notification that a communication session is initiated in a network, the notification including a session notification application programming interface (API) that includes a value for an attribute of the communication session;

ascertaining that a client device is connected to the network based on the attribute of the communication session received as part of the session notification API;

applying the attribute of the communication session received as part of the session notification API to a network policy for the network to specify a parameter for the communication session;

configuring a session configuration application programming interface (API) with the parameter for the communication session by applying the value for the attribute included in the session notification API to the network policy;

generating a configuration event that includes the session configuration API configured with the parameter for the communication session; and

communicating the configuration event to the client device.

14. The method as recited in claim 13, wherein the parameter for the communication session specifies a wireless behavior to be applied by the client device.

15. The method as recited in claim 13 or 14, further comprising: receiving an indication of a change in the communication session; generating a reconfiguration event that includes at least one of a change to the parameter for the communication session or a change to a different parameter for the communication session; and communicating the reconfiguration event to the client device.

16. The method as recited in claim 15, wherein the indication of the change includes an indication of a problem with session quality for the communication session, and wherein at least one of the change to the parameter for the communication session or the change to the different parameter is specified to increase the session quality for the communication session.

17. A method comprising:

receiving an update event including an indication of a change to a communication session in a network between a client device and at least one other device, the update event further including a session update application programming interface (API) that includes a value for an attribute of the communication session;

ascertaining that the change has occurred in the communication session based on the attribute of the communication session received as part of the session update API;

applying the attribute of the communication session received as part of the session update API to a network policy for the network to specify a parameter for the communication session;

configuring a reconfiguration application programming interface (API) with the parameter for the communication session by applying the value for the attribute included in the session update API to the network policy;

generating a reconfiguration event that includes the reconfiguration API configured with the parameter for the communication session; and

communicating the reconfiguration event to the client device.

18. The method of claim 17, wherein the indication of the change includes an indication of a problem with session quality for the communication session, and wherein the parameter for the communication session is specified to increase the session quality for the communication session.

19. The method of claim 17, wherein the indication of the change includes an indication that the client device is moving to a different location, and wherein the parameter for the communication session identifies a wireless access point that is available to provide wireless connectivity at the different location.

20. The method of claim 17, wherein the indication of the change includes an indication that the communication session is terminated, and wherein the change to the parameter for the communication session includes an indication that the communication session is terminated.

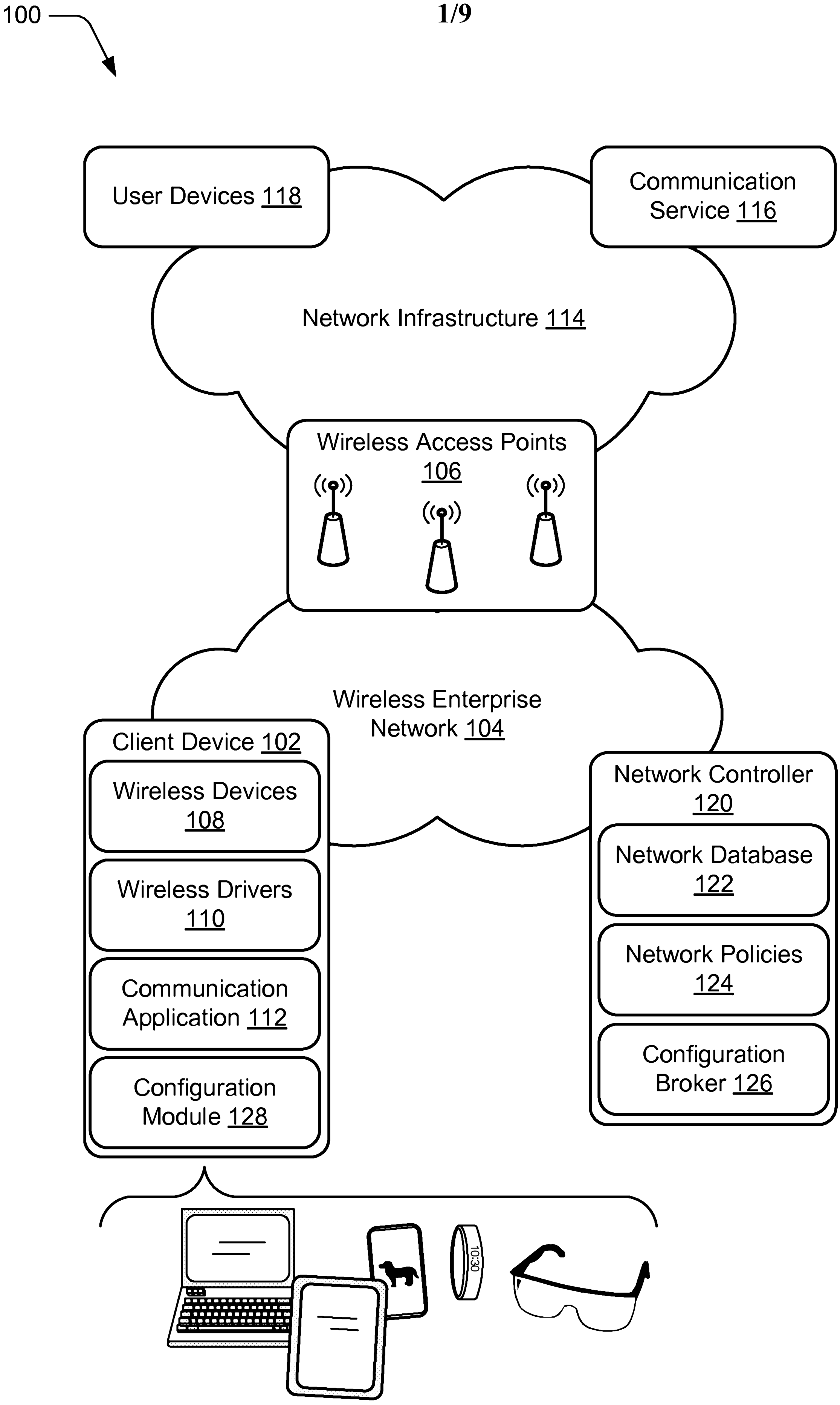
100
User Devices 118
Communication Service 116
Network Infrastructure 114
Wireless Access Points 106
Wireless Enterprise Network 104
Client Device 102
Wireless Devices 108
Wireless Drivers 110
Communication Application 112
Configuration Module 128
Network Controller 120
Network Database 122
Network Policies 124
Configuration Broker 126
10:30

Fig. 1

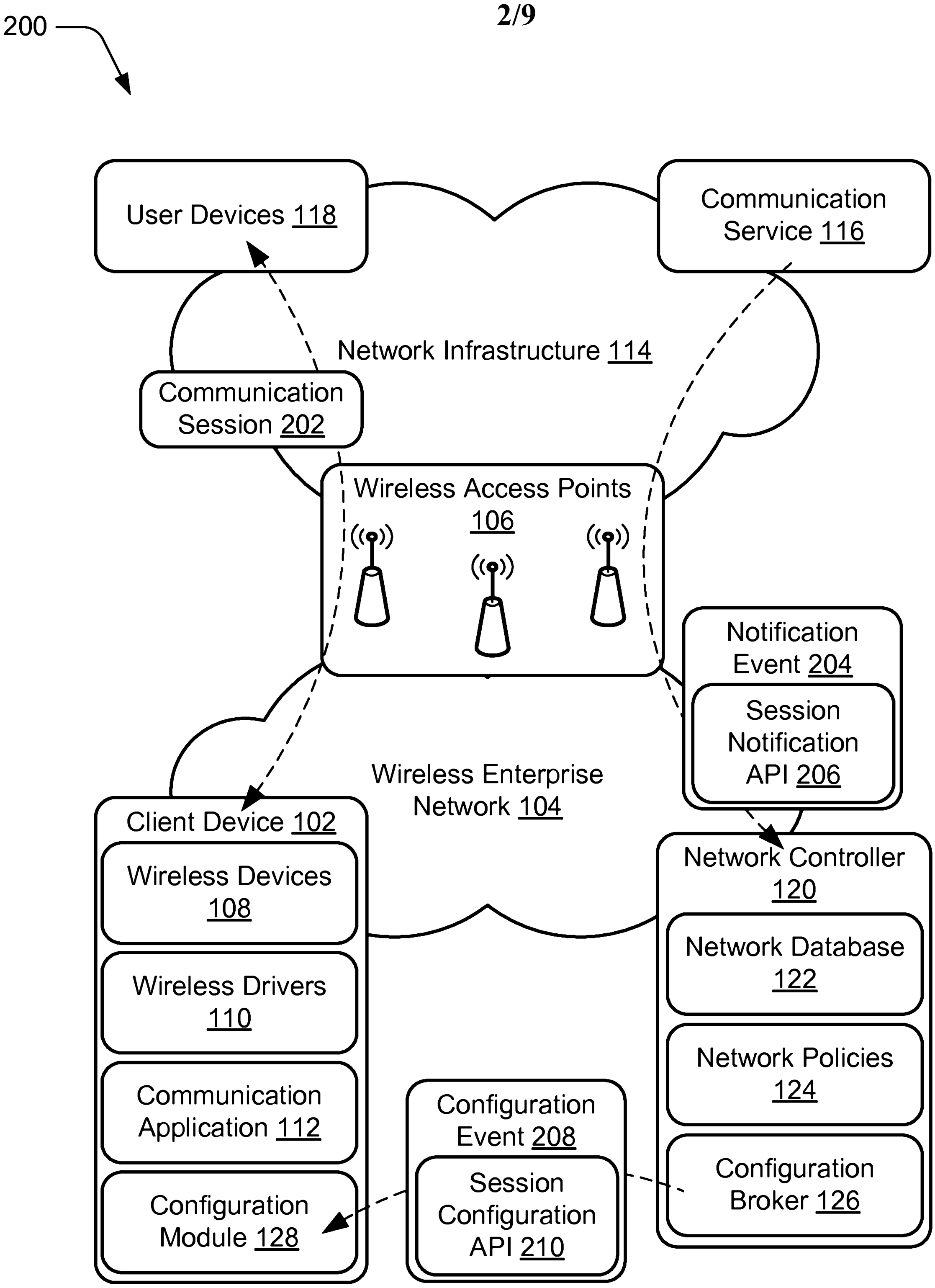
200
User Devices 118
Communication Service 116
Network Infrastructure 114
Communication Session 202
Wireless Access Points 106
Notification Event 204
Session Notification API 206
Wireless Enterprise Network 104
Client Device 102
Wireless Devices 108
Wireless Drivers 110
Communication Application 112
Configuration Module 128
Network Controller 120
Network Database 122
Network Policies 124
Configuration Broker 126
Configuration Event 208
Session Configuration API 210

Fig. 2

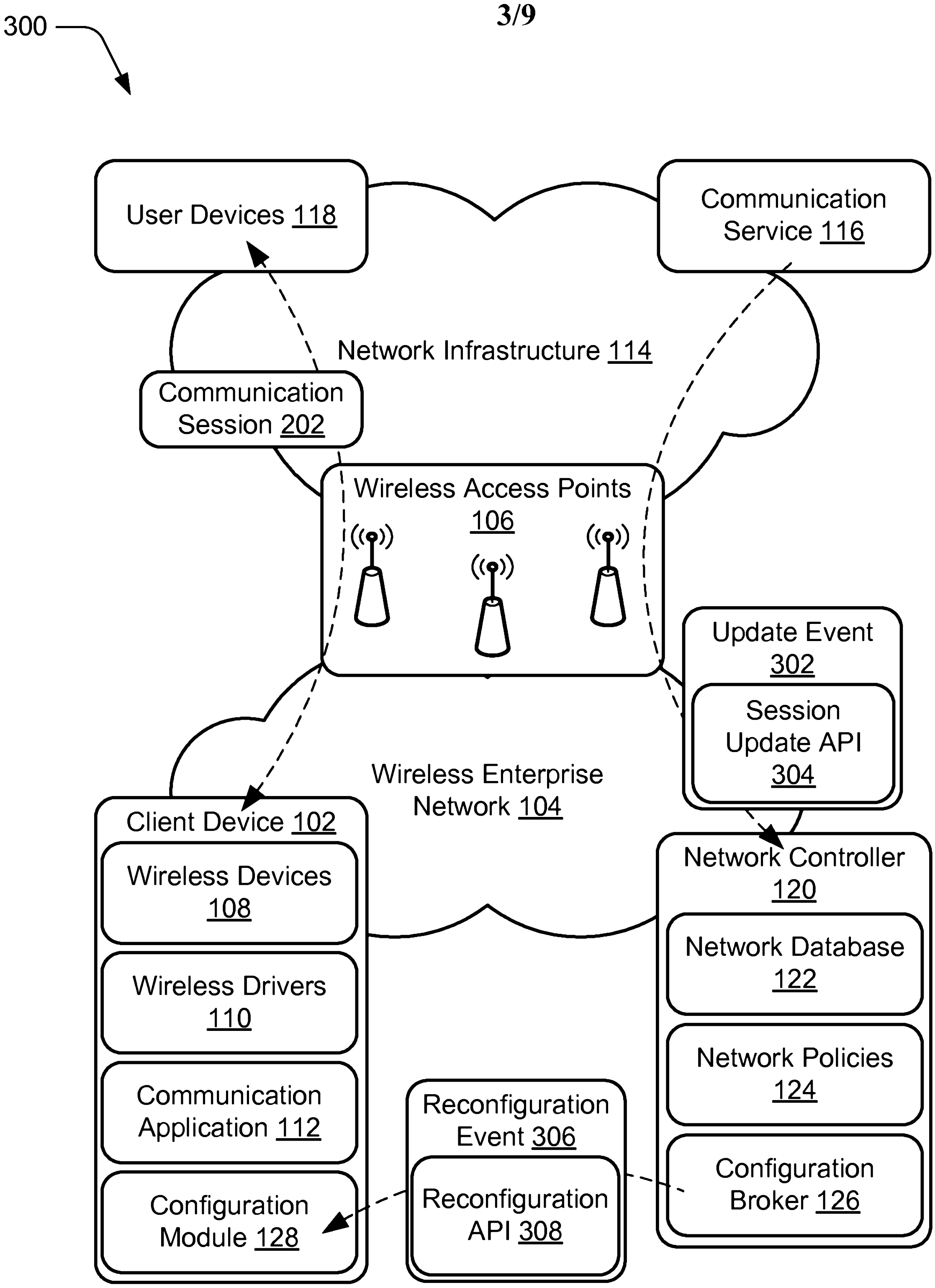
300
User Devices 118
Communication Service 116
Network Infrastructure 114
Communication Session 202
Wireless Access Points 106
Update Event 302
Session Update API 304
Wireless Enterprise Network 104
Client Device 102
Wireless Devices 108
Wireless Drivers 110
Communication Application 112
Configuration Module 128
Network Controller 120
Network Database 122
Network Policies 124
Configuration Broker 126
Reconfiguration Event 306
Reconfiguration API 308

Fig. 3

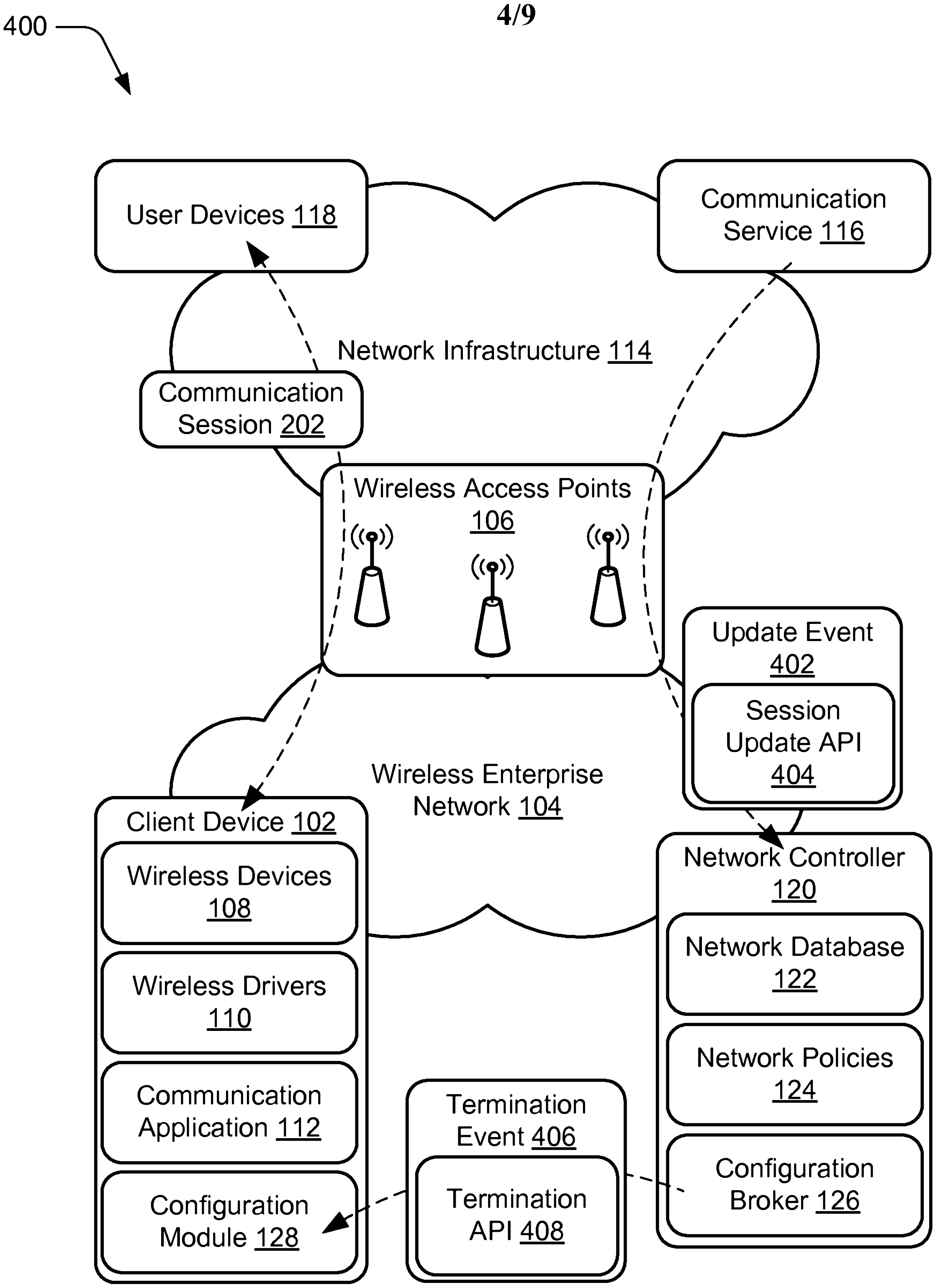
400
User Devices 118
Communication Service 116
Network Infrastructure 114
Communication Session 202
Wireless Access Points 106
Update Event 402
Session Update API 404
Wireless Enterprise Network 104
Client Device 102
Wireless Devices 108
Wireless Drivers 110
Communication Application 112
Configuration Module 128
Network Controller 120
Network Database 122
Network Policies 124
Configuration Broker 126
Termination Event 406
Termination API 408

Fig. 4

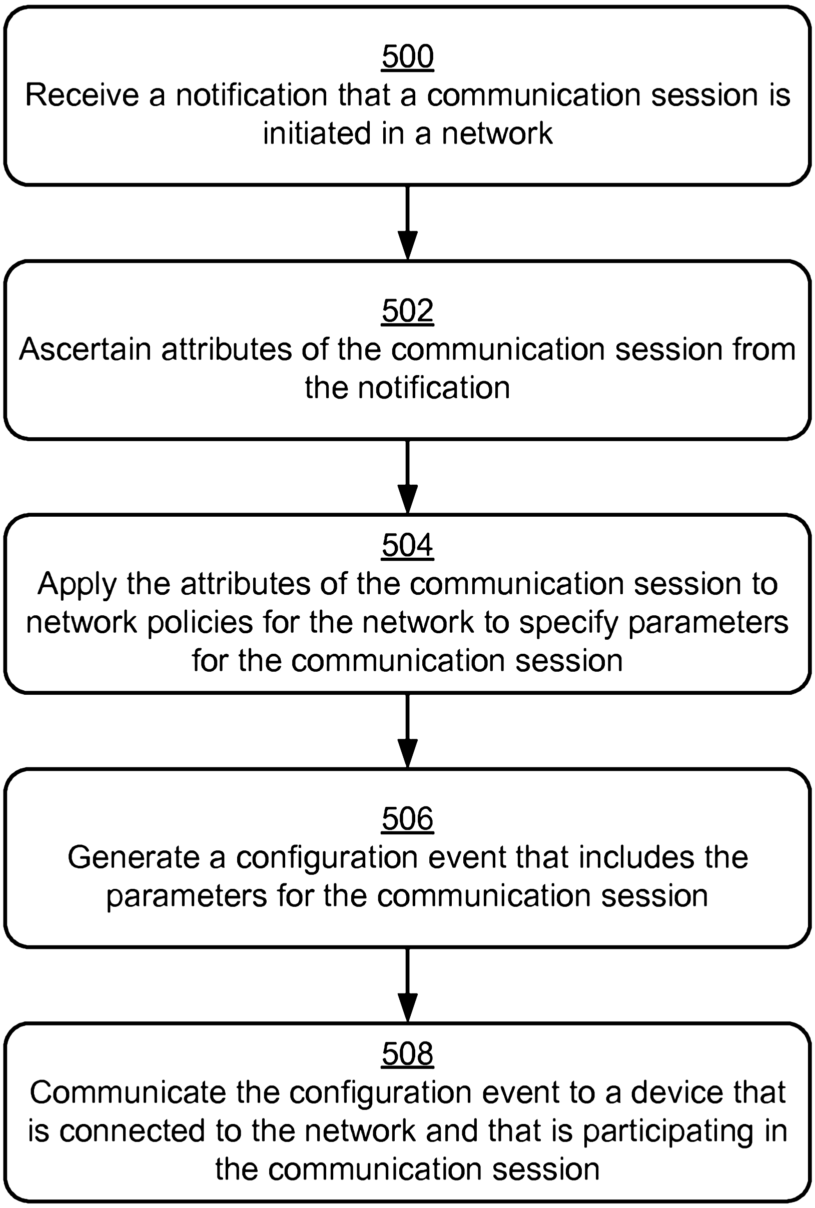
500
Receive a notification that a communication session is initiated in a network
502
Ascertain attributes of the communication session from the notification
504
Apply the attributes of the communication session to network policies for the network to specify parameters for the communication session
506
Generate a configuration event that includes the parameters for the communication session
508
Communicate the configuration event to a device that is connected to the network and that is participating in the communication session

Fig. 5

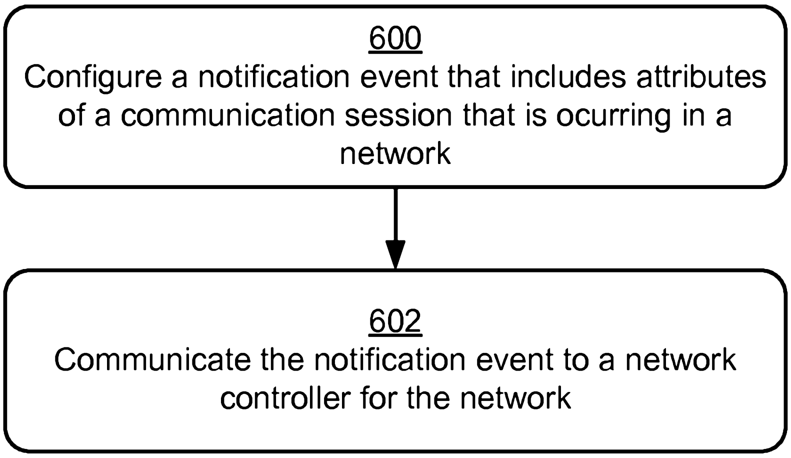
600
Configure a notification event that includes attributes of a communication session that is ocurring in a network
602
Communicate the notification event to a network controller for the network

Fig. 6

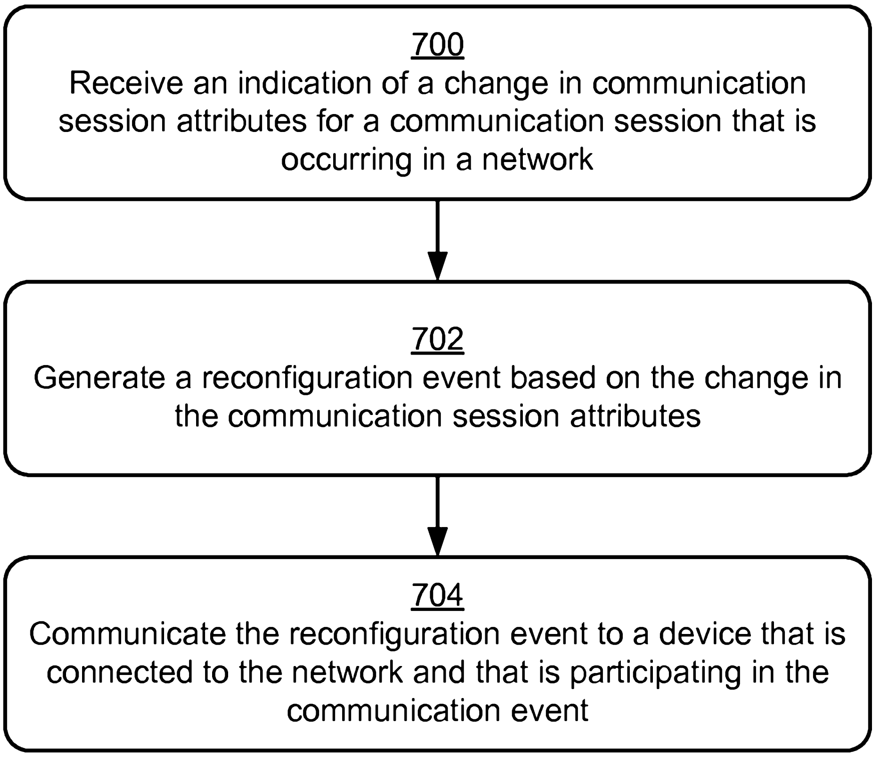
700
Receive an indication of a change in communication session attributes for a communication session that is occurring in a network
702
Generate a reconfiguration event based on the change in the communication session attributes
704
Communicate the reconfiguration event to a device that is connected to the network and that is participating in the communication event

Fig. 7

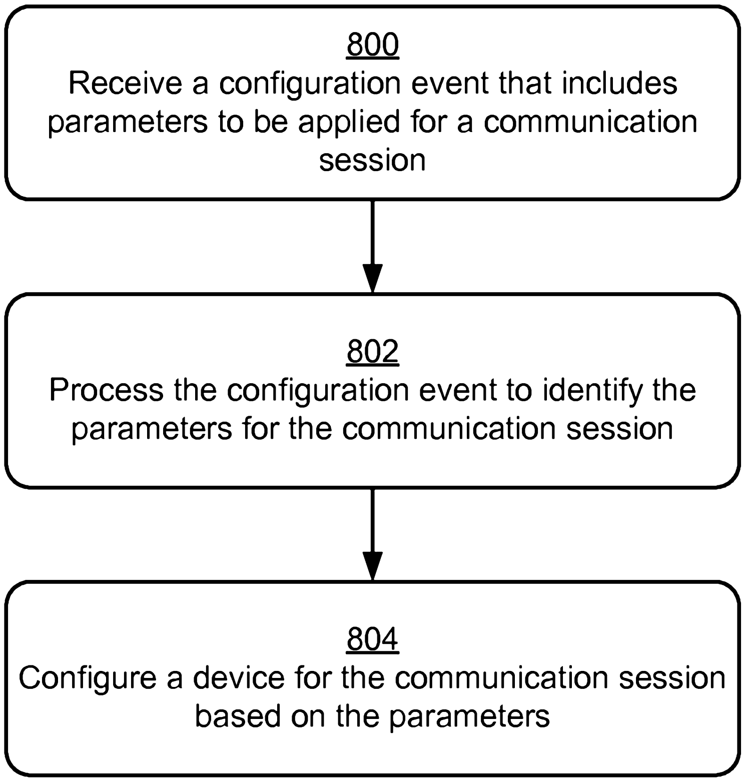
800
Receive a configuration event that includes parameters to be applied for a communication session
802
Process the configuration event to identify the parameters for the communication session
804
Configure a device for the communication session based on the parameters

Fig. 8

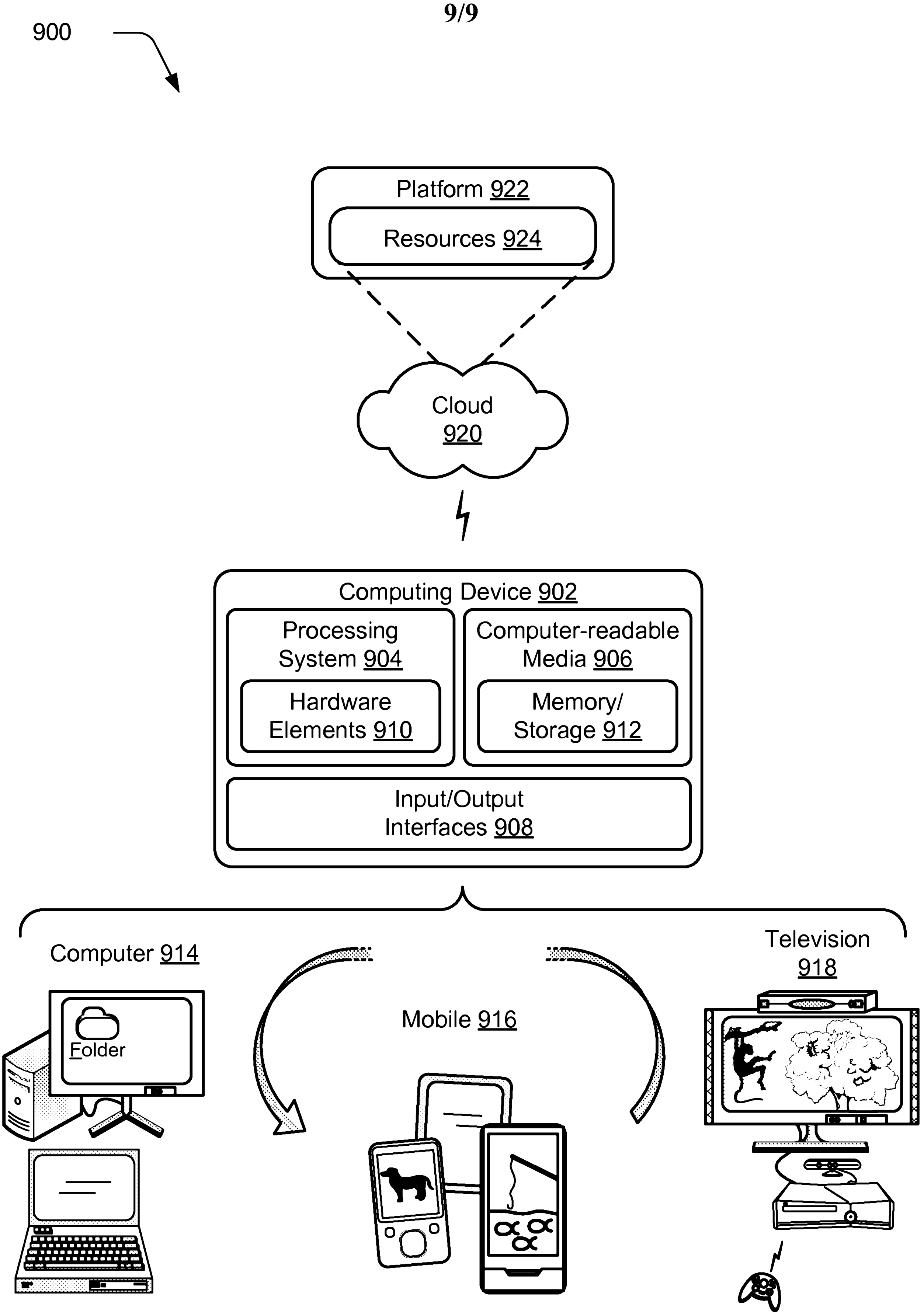
900
Platform 922
Resources 924
Cloud
920
Computing Device 902
Processing
System 904
Hardware
Elements 910
Computer-readable
Media 906
Memory/
Storage 912
Input/Output
Interfaces 908
Computer 914
Folder
Mobile 916
Television
918

Fig. 9