



(51) International Patent Classification:  
*H04W 76/02* (2009.01)

(21) International Application Number:  
PCT/IB2013/000390

(22) International Filing Date:  
8 January 2013 (08.01.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:  
61/584,774 9 January 2012 (09.01.2012) US  
13/735,458 7 January 2013 (07.01.2013) US

(71) Applicant (for all designated States except US): **MARVELL WORLD TRADE LTD.** [BB/BB]; L'Horizon,  
Gunsite Road, Brittons Hill, St. Michael, BB14027 (BB).

(72) Inventor; and

(71) Applicant : **BIJWE, Sagar, D.** [IN/IN]; 30 Maratha  
Colony, Gopal Nagar, Amravati, Maharashtra (IN).

(74) Agent: **BREEN III, William, J.**; Wolfe-SBMC, 601 W.  
Main Avenue, Suite 1300, Spokane, WA 99201 (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, IS, JP, KE, KG, KM, KN, KP,  
KR, KZ, LA, LC, LK, LR, LS, LT, 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, 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, SZ, 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,  
ML, MR, NE, SN, TD, TG).

Published:

— without international search report and to be republished  
upon receipt of that report (Rule 48.2(g))

(54) Title: METHODS AND APPARATUS FOR ESTABLISHING A TUNNELED DIRECT LINK SETUP (TDLS) SESSION  
BETWEEN DEVICES IN A WIRELESS NETWORK

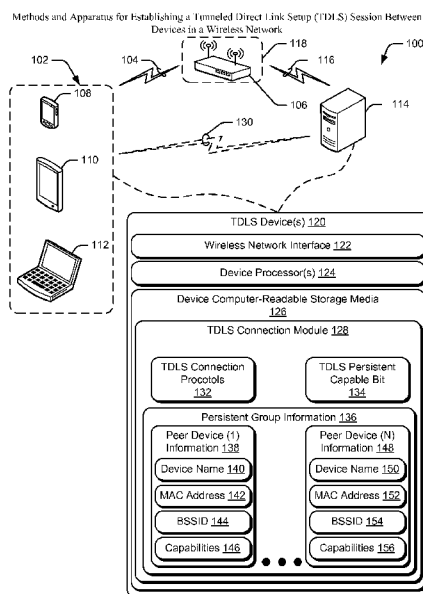


Fig. 1

(57) Abstract: The present disclosure describes techniques for establishing and reestablishing tunneled direct link setup (TDLS) sessions. In some aspects a TDLS persistent capable bit is stored at a computing device to indicate whether the computing device is capable of storing persistent group information associated with TDLS sessions. Persistent group information is also stored at the computing device to assist the computing device in reestablishing TDLS sessions and directing reestablishment of TDLS sessions.

**METHODS AND APPARATUS FOR ESTABLISHING A TUNNELED DIRECT LINK SETUP (TDLS)****SESSION BETWEEN DEVICES IN A WIRELESS NETWORK****CROSS-REFERENCE TO RELATED APPLICATIONS**

[0001] This present disclosure claims priority to U.S. Utility Application No. 13/735,458, filed on January 7, 2013 which in turn claims priority to Provisional Patent Application No. 61/584,774, filed on January 9, 2012, the disclosure of which is incorporated herein by reference.

**BACKGROUND**

[0002] The background description provided herein is for the purpose of generally presenting the context of the disclosure. Work of the presently named inventors, to the extent it is described in this background section, as well as aspects of the description that may not otherwise qualify as prior art at the time of filing, are neither expressly nor impliedly admitted as prior art against the present disclosure.

[0003] A tunneled direct link setup (TDLS), as defined by IEEE Standard 802.11z and related IEEE Standard 802.11mb, may be used to support wireless communication. A TDLS session, for instance, allows computing devices to exchange data or obtain services directly from other computing devices. A WLAN is initially used to setup the TDLS session, and the computing devices in a TDLS session are connected to the same wireless access point.

[0004] However, each time a TDLS session is setup, a TDLS discovery operation and TDLS setup operation are involved. Further, information about the computing devices, such

as connection information and capabilities, is lost each time a TDLS session is lost or terminated using conventional techniques. Thus, TDLS sessions setup using conventional techniques are typically slow and computationally expensive to setup.

### SUMMARY

[0005] This summary is provided to introduce subject matter that is further described below in the Detailed Description and Drawings. Accordingly, this Summary should not be considered to describe essential features nor used to limit the scope of the claimed subject matter.

[0006] In one aspect, the present disclosure describes a method implemented to send a TDLS discovery request from a TDLS device and to receive a TDLS discovery response at the TDLS device. The method is further implemented to send a TDLS setup request from the TDLS device, receive a TDLS setup response at the TDLS device, and send a TDLS setup confirmation from the TDLS device. Finally, the method is implemented to establish a TDLS session and store persistent information from the TDLS setup response on the TDLS device.

[0007] The method is further implemented to send, from a first TDLS device to a second TDLS device, a TDLS setup request based at least in part on persistent information about the second TDLS device, receive a TDLS setup response at the first TDLS device from the second TDLS device, and send a TDLS setup confirmation from the first TDLS device to the second TDLS device. The method is further implemented to reestablish a TDLS session between the first TDLS device and the second TDLS device without a TDLS discovery request and TDLS discovery response.

[0008] A system-on-chip (SoC) includes a transmitter configured to transmit via a wireless medium. The SoC also includes a receiver configured to receive via the wireless medium. Finally, the SoC includes a TDLS connection module configured to transmit, via the transmitter, a TDLS setup request based at least in part on persistent information about a member device of a TDLS group. The TDLS connection module is further configured to receive, via the receiver, a TDLS setup response from the member device. The TDLS connection module is also configured to transmit, via the transmitter, a TDLS setup confirmation to the member device. Lastly, the TDLS connection module is configured to establish, via the wireless medium, a TDLS session with the member device.

[0009] The details of one or more implementations are set forth in the accompanying drawings and the description below. Other features and advantages will be apparent from the description and drawings

#### **BRIEF DESCRIPTION OF THE DRAWINGS**

[0010] The details of one or more implementations are set forth in the accompanying figures and the detailed description below. In the figures, the left-most digit of a reference number identifies the figure in which the reference number first appears. The use of the same reference numbers in different instances in the description and the figures indicate like elements.

[0011] Fig. 1 illustrates an operating environment having wireless devices in accordance with one or more aspects.

[0012] Fig. 2 illustrates an example of devices of Fig. 1 setting up a TDLS session in accordance with one or more aspects.

[0013] Fig. 3 illustrates a method of establishing a TDLS session and storing information about computing devices in the TDLS session.

[0014] Fig. 4 illustrates a method of reestablishing a TDLS session based on persistent information.

[0015] Fig. 5 illustrates a method of selecting a computing device to reestablish a TDLS session with and reestablishing the TDLS session.

[0016] Fig. 6 illustrates a computing device with abstracted networking layers.

[0017] Fig. 7 illustrates a SoC environment for implementing aspects of the techniques described herein.

#### **DETAILED DESCRIPTION**

[0018] Conventional techniques for establishing a TDLS session fail to store information about the computing devices that are connected. This leads to inefficiencies in establishing TDLS sessions, namely an inability to select computing devices with which to establish a TDLS session, redundant setup operations, and so on. This disclosure describes techniques of reinvoking or reestablishing TDLS sessions based on information that is persistently stored (e.g. persistent group information) about computing devices. In one implementation, the persistent group information is stored at a particular device subsequent to the establishment of a TDLS session. The persistent group information can be used to, among other operations, direct a computing device to establish a TDLS session with another computing device that has a desired capability, attempt to make a computing

device connection with a wireless access point to enable a TDLS session setup, and otherwise manage storage of the persistent group information.

**[0019]** In an example use scenario, a user of a smart phone may come home from work and desire access to music stored at a server. With current TDLS architecture, the user experiences a delay in accessing their music each time a TDLS session is established with the server. With the TDLS architecture disclosed below with reference to Figs. 1-7, a user of the mobile phone may connect with the server with significantly reduced delay when reestablishing the session.

**[0020]** Additionally, the mobile phone may direct setup of TDLS sessions based on a user's desired actions, rather than performing TDLS discovery on all the devices in range, performing TDLS setup with all the devices in range, and then searching through the devices to find a device with the desired capabilities. In this way, if the user desired to access a separate server with personal videos and pictures stored therein, the mobile phone could direct the TDLS setup to the server with the personal videos and pictures, rather than the music server. Therefore, a user would be able to switch between listening to music and posting information to social media sites with increased efficiency. Further, if a mobile phone had previously connected with a server, once the mobile phone came within range of the server, the server could initiate reestablishment of a TDLS session with the mobile phone. Thus, memory could be saved on the mobile phone by allowing the server to store the TDLS persistent group information.

**[0021]** Also, devices in a TDLS session may be required to connect to the same access point. In another example use scenario, a mobile phone user who is walking through

a mall may have the option of connecting with multiple access points, such as access points associated with advertising servers owned by the various retail outlets. With current TDLS architecture, the mobile phone would be unable to browse available advertisements on the available servers, since each store would likely have its own access point. With the TDLS architecture disclosed below, a mobile phone user would be able to seek out access points to interact with different stores, such as to view promotions associated with different retail outlets.

[0022] Further, a user may desire to share persistent information with multiple devices. For example, a mobile phone may share persistent information with a tablet device if a user authenticates a user account on both the mobile phone and the tablet. Alternatively, a user may desire to share persistent information with another user. For example, a friend of the user with the music server may visit. Rather than going through TDLS discovery to find the servers in the original user's home, the original user may share persistent information with the friend to reduce setup time and allow directed TDLS session establishment. The sharing may utilize public action frames.

[0023] The following discussion describes an operating environment, techniques that may be employed in the operating environment, and a SoC in which components of the operating environment can be embodied. In the discussion below, reference is made to the operating environment by way of example only.

**[0024] Operating Environment**

Fig. 1 illustrates an example operating environment 100 having mobile wireless devices 102, each of which are capable of communicating data, packets, and/or frames over a

wireless connection 104, such as a wireless-local-area network (WLAN) enabled by wireless access point 106. Mobile wireless devices 102 include smart-phone 108, tablet computer 110, and laptop computer 112. Although not shown, other configurations of mobile wireless devices 102 are also contemplated such as a mobile-internet device (MID), mobile gaming console, electronic readers or books (e-readers or e-books), and so on. Non-mobile wireless devices, such as a desktop computer, server, internet protocol enabled televisions (IP TVs), and so on, are also included in the example operating environment 100. The non-mobile wireless devices, such as server 114, are capable of communicating data, packets, and/or frames over a network connection 116, such as a WLAN. Gateway device 118 includes wireless access point 106.

**[0025]** The mobile wireless devices 102 and the non-mobile wireless devices, such as server 114, are TDLS devices 120. Each TDLS device 120 includes a wireless network interface 122 for providing a wireless interface to handle various communication protocols, such as for example IEEE 802.11-2007, IEEE 802.11n, IEEE 802.11z, 802.11mb, and the like. For example, a mobile wireless device 102 can establish a TDLS session via a data-link layer when implementing the IEEE 802.11z protocol. The data-link layer includes the media access control (MAC) sub-layer. Wireless network interface may include a combined transmitter and receiver (shown) or a separate transmitter and receiver (not shown) and may be hardware combined with or separate from firmware or software.

**[0026]** TDLS devices 120 also include one or more processors 124, computer-readable storage media 126 (CRM 126), and TDLS connection module 128, which, in one implementation, is embodied on CRM 126. Processor 124 can be any suitable type of

processor, either single-core or multi-core, for executing instructions or code associated with applications and/or an operating system of a mobile wireless device 102. Processor 124 may be constructed with or fabricated from any suitable material such as silicon or other semiconductors. CRM 126 may include any suitable memory or storage device such as random-access memory (RAM), static RAM (SRAM), non-volatile RAM (NVRAM), read-only memory (ROM), or Flash memory useful to store data of applications and/or an operating system of the wireless device 102. Further, the CRM 126 may be differentiated from and therefore does not include a signal medium and thus signals per se. Example implementation of the TDLS connection module 128 is described below.

[0027] TDLS devices 120 may be configured as client devices in a wireless network having an infrastructure-based topology, such as when connecting to gateway device 118. Alternately or additionally, TDLS devices 120 may be configured to enable communication with other TDLS devices. In some cases, a mobile wireless device 102 may concurrently associate with gateway device 118 and establish a TDLS session with a TDLS device 120, such as server 114 to the mobile wireless device 102.

[0028] Gateway device 118 is illustrated as including a wireless access point 106 (i.e., access point 106). Other gateway devices 118 contemplated include various networking devices, such as routers, mobile hotspots, wireless repeaters, wireless devices configured to share a wireless connection, and so on. Gateway device 118 may provide access to resources, services (e.g., capabilities or functions), the Internet, or other networks communicatively coupled with gateway device 118. For example, smart phone 108 is able to access the Internet when connected wirelessly to gateway device 118.

[0029] Gateway device 118 may also include a wireless transceiver, which provides a wireless interface to handle various communication protocols, such as those mentioned above and elsewhere herein. Wireless transceiver may be implemented as a single transceiver or a separate transmitter and receiver, and may be hardware combined with or separate from firmware or software. A network interface of gateway device 118 enables communication with other networks, which is described in greater detail below. Gateway device 118 also includes one or more micro-processors, and gateway computer-readable storage media. Gateway CRM may include any suitable memory or storage device such as static RAM (SRAM), ROM, or Flash memory useful to store data of applications and/or an operating system of the gateway device 118.

[0030] A mobile wireless device 102 may be connected to server 114 through a TDLS session 130, which is represented through use of a phantom arrow in the figure. As illustrated, any TDLS device 120 may be connected to any other TDLS device through a TDLS session 130 after TDLS setup, as discussed below in relation to Figs. 1-5.

[0031] CRM 126 includes a TDLS connection module 128 that is configured to enable establishment or reestablishment of TDLS sessions between TDLS devices 120, such as between a mobile wireless device 102 and a server 114. TDLS connection protocols 132 are representative of the operations involved to establish or reestablish a TDLS session. TDLS persistent capable bit 134 may be used as an indication of whether the TDLS device is capable of storing persistent information associated with TDLS sessions.

[0032] Persistent group information 136 represents information that is persistently stored and is associated with TDLS device (1) through TDLS device (N). TDLS devices (1)

through TDLS device (N) are other TDLS devices with which the TDLS device 120 has previously established a TDLS session. For example, server 114 may store persistent group information 136 associated with mobile wireless devices 108-112. In addition, smart phone 108 may store persistent group information 136 associated with server 114. Persistent group information 136 is stored in a persistent manner, such that if a TDLS session is terminated, the persistent group information 136 is maintained on one or both of the TDLS devices 120 that were connected through the TDLS session. Thus, persistent group information 136 may be stored upon establishment of a TDLS session, or at any time during the TDLS session.

**[0033]** The persistent group information 136 may include a variety of different types of information. For example, TDLS device (1) information 138 may include a device name 140 of TDLS device (1). The device name 140 may be customizable, and may be set by a user or administrator of TDLS device (1). TDLS device (1) information 138 may also include a MAC address 142 of TDLS device (1). TDLS device (1) information 138 may further include a basic service set identifier (BSSID) 144. The BSSID 144 may include or be set to the MAC address of the access point 106 that was used to establish the TDLS session between TDLS device 120 and TDLS device (1). TDLS device (1) information 138 may also include capabilities 144 associated with the applications present on TDLS device (1). Capabilities 144 may also be associated with hardware or other characteristics of TDLS device (1). Similar information to TDLS device (1) information 138 may be stored for each TDLS device with which TDLS device 120 has previously established a TDLS session, for example TDLS device (N) information 146.

**[0034] Techniques of Establishing TDLS Sessions**

**[0035]** Fig. 2 illustrates an example of smart phone 108 setting up a TDLS session with server 116. Device environment 200 includes smart phone 108, server 114, and access point 106 of Fig. 1.

**[0036]** When smart phone 108 is connected to access point 106, smart phone 108 may be configured to transmit an open access TDLS discovery request 202. The TDLS discovery request includes a TDLS persistent capable bit 134, which is an indication of whether the TDLS device is capable of storing persistent information associated with TDLS sessions. Also, the TDLS persistent capable bit may be included within existing TDLS discovery request communication protocols. For example, the TDLS persistent capable bit may be included as the unassigned bit within the current TDLS discovery request message architecture. Access point 106 may receive TDLS discovery request 202, and transmit TDLS discovery request 202 to server 114. Server 114 may then receive the TDLS discovery request 202 and send TDLS discovery response 204 to smart phone 108 through access point 106. Alternatively, server 114 may send TDLS discovery response 204 directly to smart phone 108 through a unicast public action frame. Further, server 114 may send TDLS discovery response to smart phone 108 in an unsolicited manner, where the smart phone 108 does not send a TDLS discovery request 202. TDLS discovery response 204 may include a TDLS persistent capable bit 134 and other information necessary to establish a TDLS session. For example, TDLS discovery response 204 may include the device name of server 114, the MAC address of server 114, and the BSSID of server 114.

[0037] Upon receipt of the TDLS discovery response 204, smart phone 108 initiates a TDLS setup request 206. The TDLS setup request 206 may be based at least in part on the information about server 114, and uses this information to pass through access point 106 to server 114. Server 114 responds to TDLS setup request 206 with TDLS setup response 208, which passes through access point 106 and is received at smart phone 108. Smart phone 108 then sends TDLS setup confirmation 210 to server 114 through access point 106. A TDLS session 212 may be established at smart phone 104 any time after smart phone 108 sends TDLS setup confirmation 210. In addition, a TDLS session may be established at server 114 any time after server 114 receives TDLS setup confirmation 210.

**[0038] Techniques of Establishing and Reestablishing a TDLS Session**

[0039] The following discussion describes techniques for establishing and reestablishing or reinvoking a TDLS session based on persistent group information. These techniques can be implemented using the previously described environments or techniques, such as TDLS connection module 128 of Fig. 1 embodied on a TDLS device 120 and the method described in Fig. 2. These techniques include methods illustrated in Figs. 3, 4, and 5, each of which is shown as a set of operations performed by one or more entities. These methods are not necessarily limited to the orders shown for performing the operations. Further, these methods may be used in conjunction with one another, in whole or in part, whether performed by the same entity, separate entities, or any combination thereof. In portions of the following discussion, reference will be made to operating environment 100 of Fig. 1 and techniques of Fig. 2 by way of example. Such reference is not

to be taken as limited to operating environment 100 but rather as illustrative of one of a variety of examples.

[0040] Fig. 3 illustrates a method of establishing a TDLS session and storing information about TDLS devices 120 in the TDLS session. By using this method when first establishing a TDLS session, the persistent group information gathered during this method allows methods, such as those illustrated below in relation to Figs. 4 and 5, to utilize the persistent group information. The persistent group information makes reestablishing a TDLS session more efficient, less computationally costly, and allows a TDLS session to be established when one or neither of the TDLS devices is connected to the same access point or either of the TDLS devices is connected to a different access point.

[0041] At 302, a TDLS device generates or sets a TDLS persistent capable bit 134 for the TDLS device 120. This TDLS persistent capable bit 134 is an indication whether the TDLS device 120 supports storing persistent information associated with TDLS sessions. The TDLS persistent capable bit 134 could be set, for example, to 0 to signify that the TDLS device 120 is does not store persistent information associated with TDLS sessions. Conversely, the TDLS persistent capable bit 134 could be set to 1 to signify that the TDLS device 120 stores persistent information associated with TDLS sessions. The TDLS persistent capable bit 134 could be larger than a single bit, and could be set to any value to signify that the TDLS device stores or does not store persistent information associated with TDLS sessions.

[0042] At 304, the TDLS device 120 sends out an open access TDLS discovery request 202. For example, the TDLS discovery request 202 may be transmitted on a public action

frame. The TDLS discovery request 202 seeks out other TDLS devices and includes the TDLS persistent capable bit 134.

[0043] At 306, the TDLS device 120 receives a TDLS discovery response 204. The TDLS discovery response 204 comes from another TDLS device, and includes information necessary to establish a TDLS session between the TDLS device 120 and the other TDLS device. This information could include the other TDLS device's name, MAC address, BSSID, and application capabilities.

[0044] At 308, optionally, if the other TDLS device is not connected to the same access point as the TDLS device 120, the TDLS device 120 may ask the other TDLS device to connect to the same access point. TDLS sessions may require the TDLS device 120 and the other TDLS device to be connected to the same access point for TDLS setup.

[0045] At 310, the TDLS device 120 performs a TDLS setup with the other TDLS device. These operations are similar to those described above in relation to Fig. 2, specifically TDLS setup request 206, TDLS setup response 208, and TDLS setup confirmation 210.

[0046] At 312, the TDLS device 120 stores, in a location that is accessible to the TDLS device, information about the other TDLS device as persistent group information 136. This information could include the other device's name 140, MAC address 142, BSSID 144, and capabilities 146. The persistent group information 136 allows a TDLS session to be reestablished more efficiently, at a lower computational cost, and allows a TDLS session to be established when one or neither of the TDLS devices is connected to the same access point or either of the TDLS devices is connected to a different access point.

[0047] At 314, the TDLS device 120 and the other TDLS device establish a TDLS session. After the TDLS session is established, application data may be exchanged between the TDLS device 120 and the other TDLS device.

[0048] Fig. 4 illustrates a method 400 of reestablishing or reinvoking a TDLS session based on persistent group information.

[0049] At 402, a TDLS device, such as smart phone 108 of Fig. 1, requests that a TDLS session be established with another TDLS device, such as server 114 of Fig. 1.

[0050] At 404, the TDLS device 120 determines whether a TDLS session with the other TDLS device currently exists.

[0051] At 406, if a TDLS session currently exists between the TDLS device 120 and the other TDLS device, the TDLS device 120 uses application specific messages to exchange application data between the TDLS device 120 and the other TDLS device. For example, TDLS device 120 and the other TDLS device may exchange protocol messages to setup playback of audio or video files, or may exchange audio or video files.

[0052] At 408, if a TDLS session does not currently exist between the TDLS device 120 and the other TDLS device, the TDLS device 120 determines whether the TDLS device 120 is associated with a stored access point. The stored access point was previously used to setup a TDLS session between TDLS device 120 and the other TDLS device. The stored access point may be stored in persistent group information 136, and may be identified by the BSSID.

[0053] At 410, if the TDLS device 120 is not associated with the stored access point, the TDLS device 120 attempts to associate itself with the stored access point.

[0054] At 412, if the TDLS device 120 was previously associated with the stored access point or if the TDLS device 120 successfully associates itself with the stored access point, the TDLS device 120 attempts to setup a TDLS session. The attempt is similar to the operations described above in relation to Fig. 2, specifically TDLS setup request 206, TDLS setup response 208, and TDLS setup confirmation 210.

[0055] At 414, if the attempt to setup a TDLS session times out or otherwise fails, the TDLS device 120 sends out a message, for example a public action frame message, asking the other TDLS device to connect with the stored access point. The TDLS device 120 may then wait for a pre-determined time, and then attempt to setup a TDLS session 412 again. The TDLS device 120 may send out the message asking the other TDLS device to connect a predetermined number of times, including only once or more than once.

[0056] At 416, if the attempt to setup a TDLS session between the TDLS device 120 and the other TDLS device is successful, a TDLS session is established. After the TDLS session is established, the TDLS device 120 and the other TDLS device may exchange data directly, including application data.

[0057] At 418, if the TDLS device 120 fails to associate itself with the stored access point or if the TDLS setup operation times out or otherwise fails, including after asking the other TDLS device to connect a predetermined number of times, the persistent group information associated with the other TDLS device is deleted from the TDLS device 120. Upon deletion of the persistent group information, if the TDLS device attempts to establish a TDLS session with the other TDLS device at a later time, default TDLS setup procedures are utilized. Alternatively, persistent group information may be retained despite failure or time

outs.

**[0058]** Fig. 5 illustrates a method of selecting a TDLS device 120 to reestablish a TDLS session with and reestablishing the TDLS session with the selected TDLS device.

**[0059]** At 502, the TDLS device 120 selects another TDLS device from a plurality of TDLS devices with which to establish a TDLS session. The plurality of TDLS devices may be a list of TDLS devices with which the TDLS device 120 has previously established a TDLS session and with which the TDLS device 120 has persistent group information 136 on. The selection may be based on the capabilities of the selected TDLS device. These capabilities may be determined by searching the persistent group information 136. For example, referring to Fig 1, TDLS device (1) may be selected from the plurality of TDLS devices TDLS device (1) through TDLS device (N) based on capabilities 146 containing a desired function, which capabilities 156 do not contain, such as capabilities 146 containing video playback when capabilities 156 only contain audio playback.

**[0060]** At 504, the TDLS device 120 sends a TDLS setup request 206 to the selected TDLS device. The TDLS setup request 206 utilizes the persistent group information 136 associated with the selected TDLS device. By using the persistent group information 136, the TDLS setup request 206 can be sent to the selected TDLS device without performing TDLS discovery on all the TDLS devices located within range of the TDLS device.

**[0061]** At 506, optionally, if the selected TDLS device is not connected to an access point stored in the persistent group information 136, the TDLS device 120 asks the selected TDLS device to connect to the stored access point. For example, the TDLS device 120 may send a public action frame to the selected TDLS device with the BSSID of the stored access

point. An example indication that the selected TDLS device is not connected to an access point is a setup timeout.

[0062] At 508, the TDLS device 120 receives a TDLS setup response 208 from the selected TDLS device. Once the TDLS setup response 208 is received, a TDLS session may be established.

[0063] At 510, the TDLS device 120 sends a TDLS setup confirmation 210 to the selected TDLS device.

[0064] At 512, the TDLS device 120 establishes a TDLS session with the selected TDLS device. Once a TDLS session has been established, the TDLS device 120 and the other TDLS device may exchange data directly, without utilizing access point 106.

[0065] **Networking Layers of TDLS Device**

[0066] Fig. 6 illustrates an example of a TDLS device abstracted into networking layers including an application layer 602, data-link layer 604, and a physical layer 606.

[0067] TDLS device 120 can logically be divided into an application layer 602, data-link layer 604, and a physical layer 606. These layers generally correspond to the networking layers in the Open Systems Interconnection (OSI) model.

[0068] Application layer 602 includes application data, application instructions, and other information directly accessible to a user of the TDLS device.

[0069] Data-link layer 604 includes functional and procedural instructions on how to transfer data in a network. Data-link layer includes MAC sub-layer. TDLS persistent capable bit 134 and persistent group information 136 of TDLS device 120 may be stored in the data-link layer, specifically in the MAC sub-layer.

[0070] Physical layer 606 includes hardware networking technologies. Physical layer 606 may include wireless network interface 122 of TDLS device 120.

[0071] **System-on-Chip**

Fig. 7 illustrates a SoC 700, which can implement various embodiments described above. A SoC can be implemented in any suitable device, such as a video game console, IP enabled television, smart-phone, desktop computer, laptop computer, tablet computer, server, network-enabled printer, set-top box, printer, scanner, camera, picture frame, and/or any other type of device that may implement wireless connective technology.

[0072] SoC 700 can be integrated with electronic circuitry, a microprocessor, memory, input-output (I/O) logic control, communication interfaces and components, other hardware, firmware, and/or software needed to provide communicative coupling for a device, such as any of the above-listed devices. SoC 700 can also include an integrated data bus that couples the various components of the SoC 700 for data communication between the components. A wireless communication device that includes SoC 700 can also be implemented with many combinations of differing components. In some cases, these differing components may be configured to implement concepts described herein over a wireless connection or interface.

[0073] In this example, SoC 700 includes various components such as an input-output (I/O) logic control 702 (e.g., to include electronic circuitry) and a microprocessor 704 (e.g., any of a microcontroller or digital signal processor). SoC 700 also includes a memory 706, which can be any type of RAM, low-latency nonvolatile memory (e.g., Flash memory), ROM, and/or other suitable electronic data storage. SoC 700 can also include various

firmware and/or software, such as an operating system 708, which can be computer-executable instructions maintained by memory 706 and executed by microprocessor 704. SoC 700 can also include other various communication interfaces and components, communication components, other hardware, firmware, and/or software.

[0074] SoC 700 includes wireless transmitter 710, wireless receiver 712, and TDLS connection module 128 (embodied as disparate or combined components as noted above). Examples of these various components, functions, and/or entities, and their corresponding functionality, are described with reference to the respective components of the environment 100 shown in Fig. 1.

[0075] TDLS connection module 128, either independently or in combination with other entities, can be implemented as computer-executable instructions maintained by memory 606 and executed by microprocessor 604 to implement various embodiments and/or features described herein. TDLS connection module 128 may also be provided integral with other entities of the SoC 700, such as integrated with one or both of I/O logic controller 702 or any packet-based interface within SoC 700. Alternatively or additionally, TDLS connection module 128 and the other components can be implemented as hardware, firmware, fixed logic circuitry, or any combination thereof that is implemented in connection with the I/O logic control 602 and/or other signal processing and control circuits of SoC 700.

[0076] Although the subject matter has been described in language specific to structural features and/or methodological operations, it is to be understood that the

subject matter defined in the appended claims is not necessarily limited to the specific features or operations described above, including orders in which they are performed.

**CLAIMS**

What is claimed is:

**1.** A method for establishing a tunnel direct link setup (TDLS) session between a first TDLS device and a second TDLS device, the method comprising:

sending, from the first TDLS device to the second TDLS device, a TDLS discovery request;

in response to receiving, at the first TDLS device from the second TDLS device, a response to the TDLS discovery request, sending a TDLS setup request from the first TDLS device to the second TDLS device;

in response to receiving, at the first TDLS device from the second TDLS device a response to the TDLS discovery request,:

(i) sending a TDLS setup confirmation from the first TDLS device to the second TDLS device;

(ii) establishing the TDLS session between the first TDLS device and the second TDLS device; and

(iii) persistently storing, at the first TDLS device, information associated with the second TDLS device, wherein the information is persistently stored at the first TDLS subsequent to establishment of the TDLS session between the first TDLS device and the second TDLS device.

**2.** The method of claim 1, wherein the TDLS discovery request includes an indication that the first TDLS device is capable of storing persistent information associated with TDLS sessions.

**3.** The method of claim 1, wherein the persistent information comprises:  
an indication whether the first TDLS device is capable of storing persistent information associated with TDLS sessions comprising a TDLS persistent capable bit; and  
for each additional TDLS device with which the TDLS device establishes a TDLS session:

a device name of the additional TDLS device;

a MAC address of the additional TDLS device; and

a basic service set identifier (BSSID) associated with the additional TDLS device comprising a MAC address of an access point used to start the TDLS session between the TDLS device and the additional TDLS device.

**4.** The method of claim 1, further comprising discovering capabilities of applications for each additional TDLS device with which the first TDLS device establishes a TDLS session.

**5.** The method of claim 4, wherein the persistent information comprises the discovered capabilities of applications for each additional TDLS device with which the TDLS device establishes a TDLS session.

6. The method of claim 1, wherein the persistent information is stored in the MAC layer.

7. The method of claim 2, further comprising including the indication within existing TDLS discovery request communication protocols.

8. A method for reestablishing a tunnel direct link setup (TDLS) session between a first TDLS device and a second TDLS device, the method comprising:

selecting, from a plurality of TDLS devices, the second TDLS device based at least in part on persistent information stored on the first TDLS device

sending, from the first TDLS device to the selected TDLS device, a TDLS setup request based at least in part on the persistent information about the selected TDLS device;

receiving a TDLS setup response at the first TDLS device from the selected TDLS device;

sending a TDLS setup confirmation from the first TDLS device to the selected TDLS device; and

establishing the TDLS session between the first TDLS device and the selected TDLS device.

- 9.** The method of claim 8, wherein the persistent information comprises:
- an indication whether the first TDLS device is capable of storing persistent information associated with TDLS sessions comprising a TDLS persistent capable bit; and
- persistent group information.
- 10.** The method of claim 9, wherein the persistent group information comprises:
- a device name for the selected TDLS device;
- a media access control (MAC) address for the selected TDLS device; and
- a basic service set identifier (BSSID).
- 11.** The method of claim 10, wherein the BSSID comprises a MAC address for an access point that was used to connect the first TDLS device and the selected TDLS device.
- 12.** The method of claim 9, further comprising:
- attempting to initiate a connection between the selected TDLS device and an access point;
- receiving an indication that the TDLS request has timed out a predetermined number of times; and
- deleting the persistent group information.

**13.** The method of claim 8, further comprising sharing the persistent information with a third TDLS device such that the third TDLS device is enabled to establish a TDLS session with the second TDLS device without performing TDLS discovery.

**14.** The method of claim 12, wherein deleting the persistent group information comprises deleting the persistent group information associated with the second TDLS device from the first TDLS device.

**15.** The method of claim 8, wherein the device name is customizable.

**16.** The method of claim 8, further comprising attempting to initiate a connection between the first TDLS device and an access point associated with the selected TDLS device stored in the persistent information.

**17.** The method of claim 16, further comprising:

receiving an indication of failure to initiate a connection between the first TDLS device and the access point associated with the selected TDLS device stored in the persistent information; and

deleting the persistent information associated with the selected TDLS device from the first TDLS device.

**18.** A system comprising:

a transmitter configured to transmit, from a first tunneled direct link setup (TDLS) device, via a wireless medium a TDLS setup request based at least in part on persistent information about a second TDLS device of a TDLS group;

a receiver configured to receive via the wireless medium a TDLS setup response from the second TDLS device; and

a TDLS connection module configured to establish, via the wireless medium, a TDLS session between the first TDLS device and the second TDLS device.

**19.** The system of claim 18, wherein the persistent information comprises:

an indication whether the first device is capable of storing persistent information associated with TDLS sessions; and

information about the TDLS group comprising:

a device name for the second TDLS device;

a media access control (MAC) address for the second TDLS device;

a basic service set identifier (BSSID) for an access point corresponding to the second TDLS device; and

capabilities of applications stored on the second TDLS device.

**20.** The system of claim 18, wherein the TDLS connection module is further configured to:

attempt to initiate a connection between the second TDLS device and an access point;

receive a message comprising an indication that the TDLS request has timed out a predetermined number of times; and

delete the persistent information associated with the second TDLS device from the first TDLS device.

Methods and Apparatus for Establishing a Tunneled Direct Link Setup (TDLS) Session Between Devices in a Wireless Network

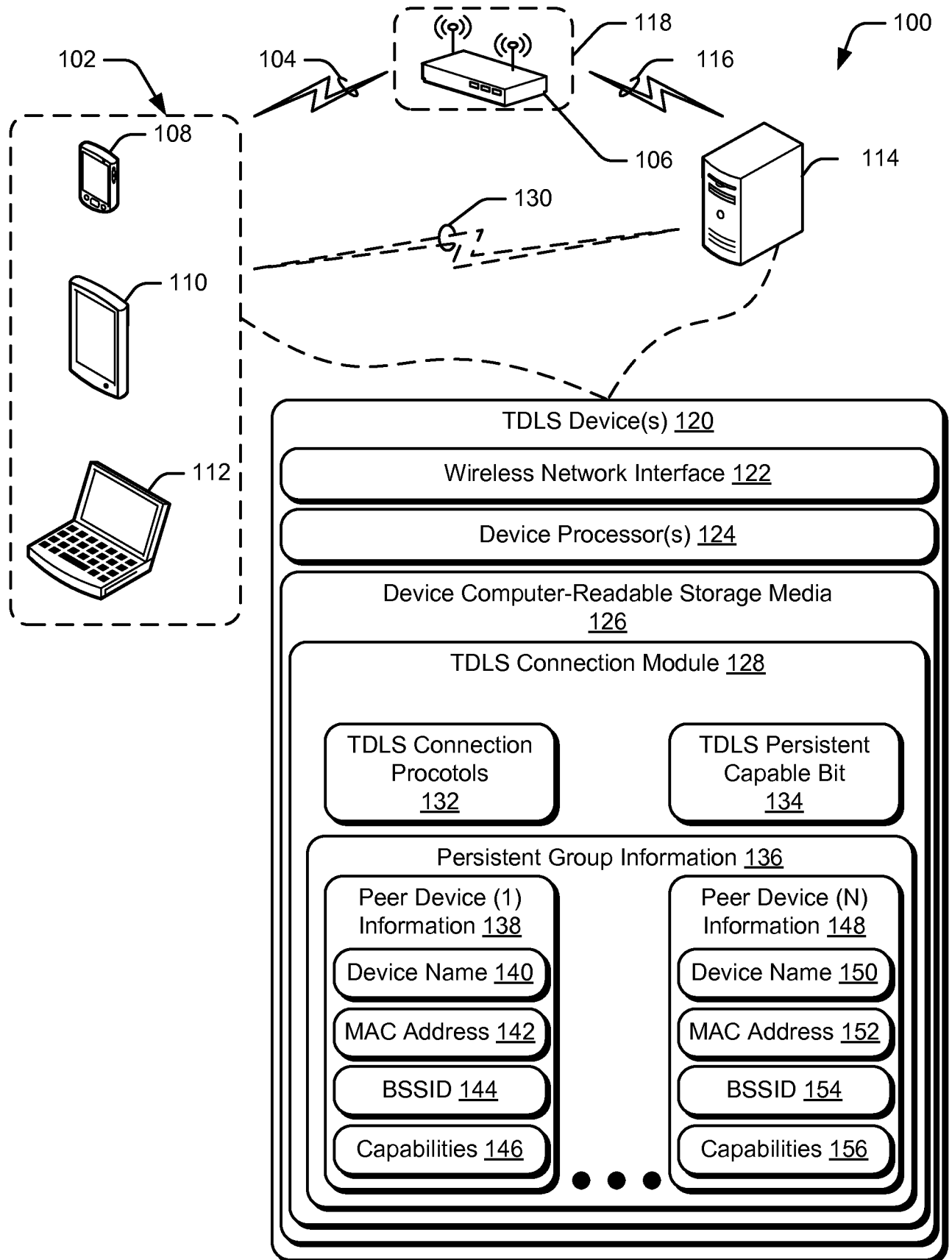


Fig. 1

## Methods and Apparatus for Establishing a Tunneled Direct Link Setup (TDLS) Session Between Devices in a Wireless Network

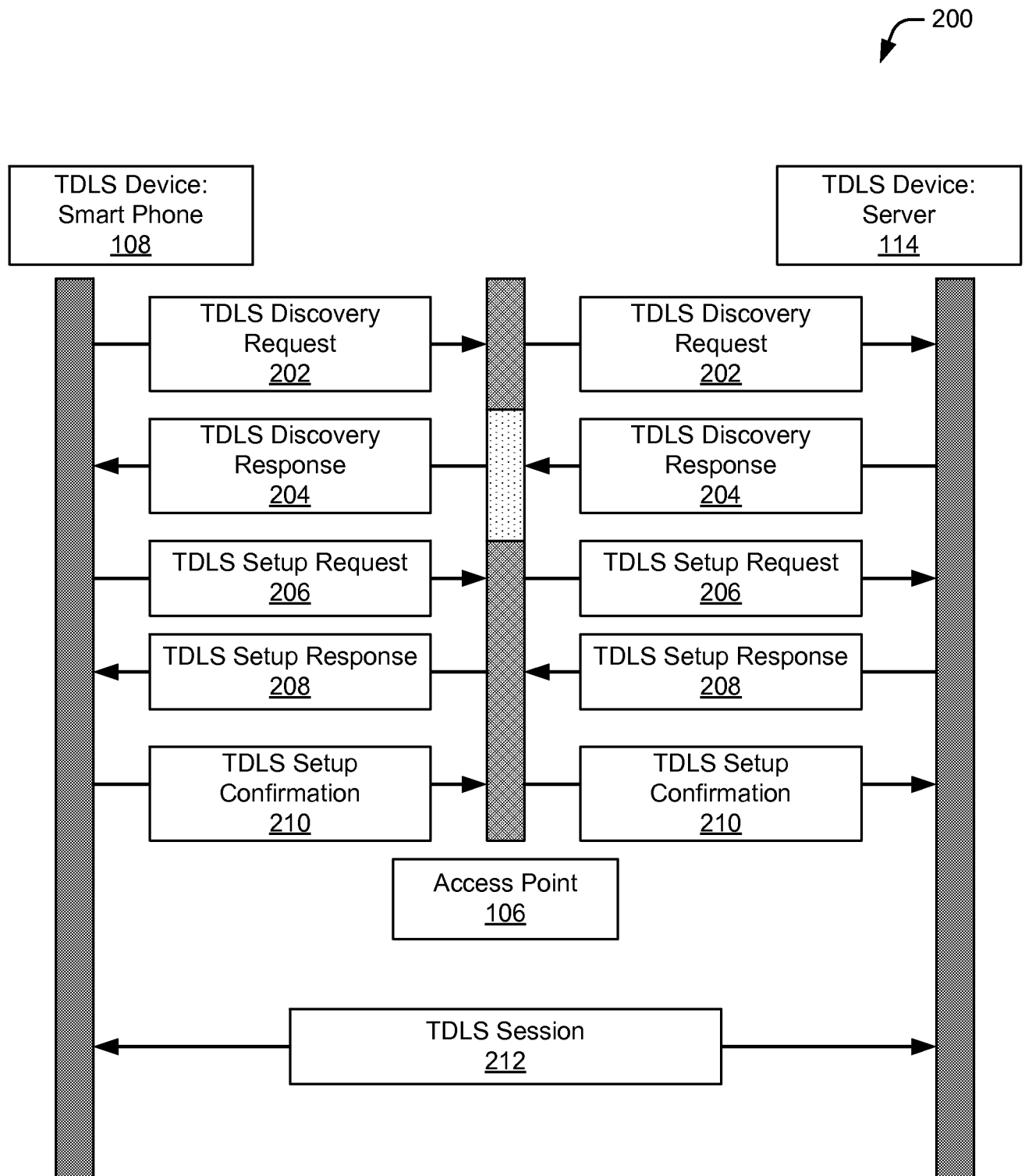


Fig. 2

## Methods and Apparatus for Establishing a Tunneled Direct Link Setup (TDLS) Session Between Devices in a Wireless Network

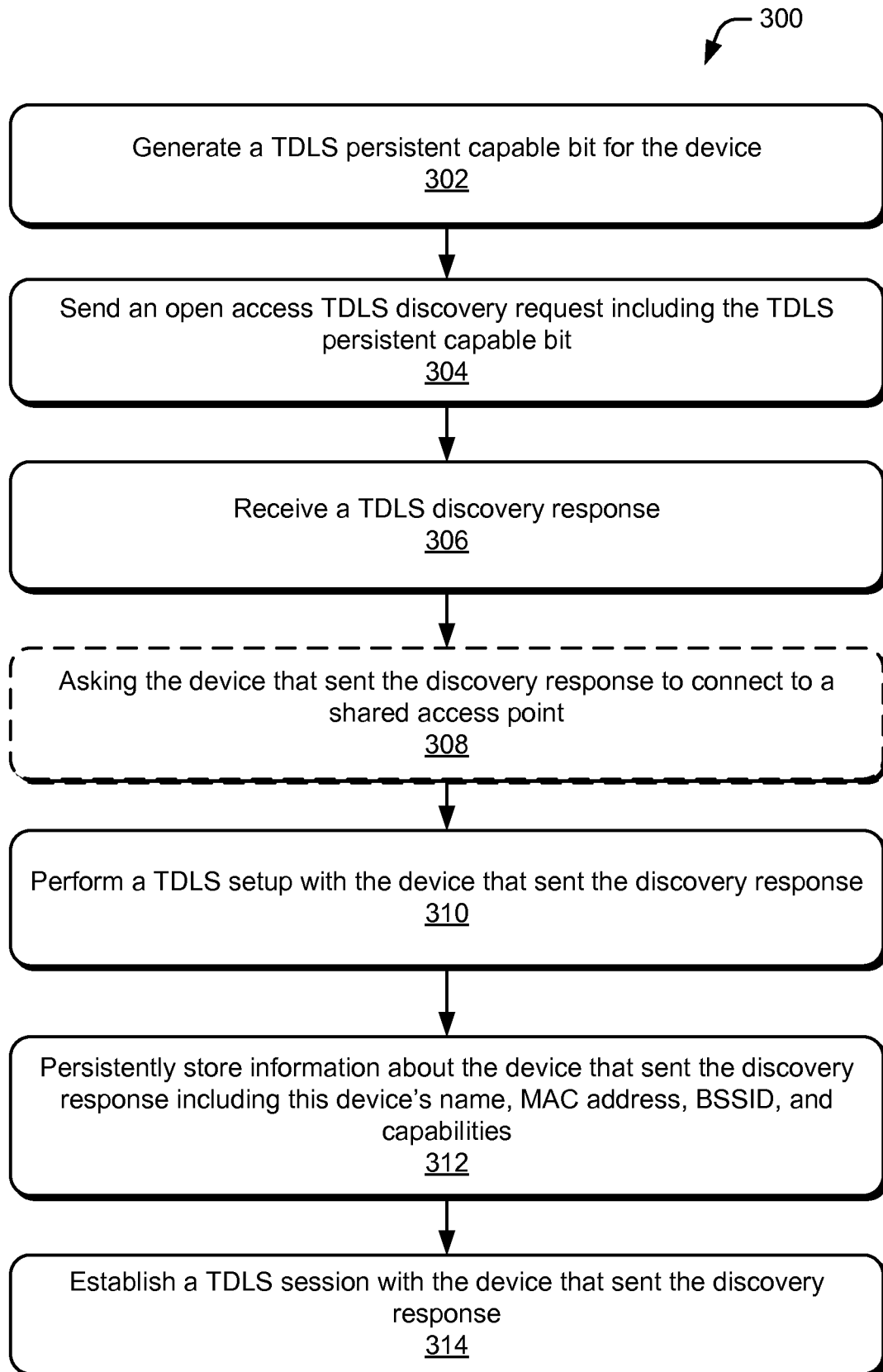


Fig. 3

## Methods and Apparatus for Establishing a Tunneled Direct Link Setup (TDLS) Session Between Devices in a Wireless Network

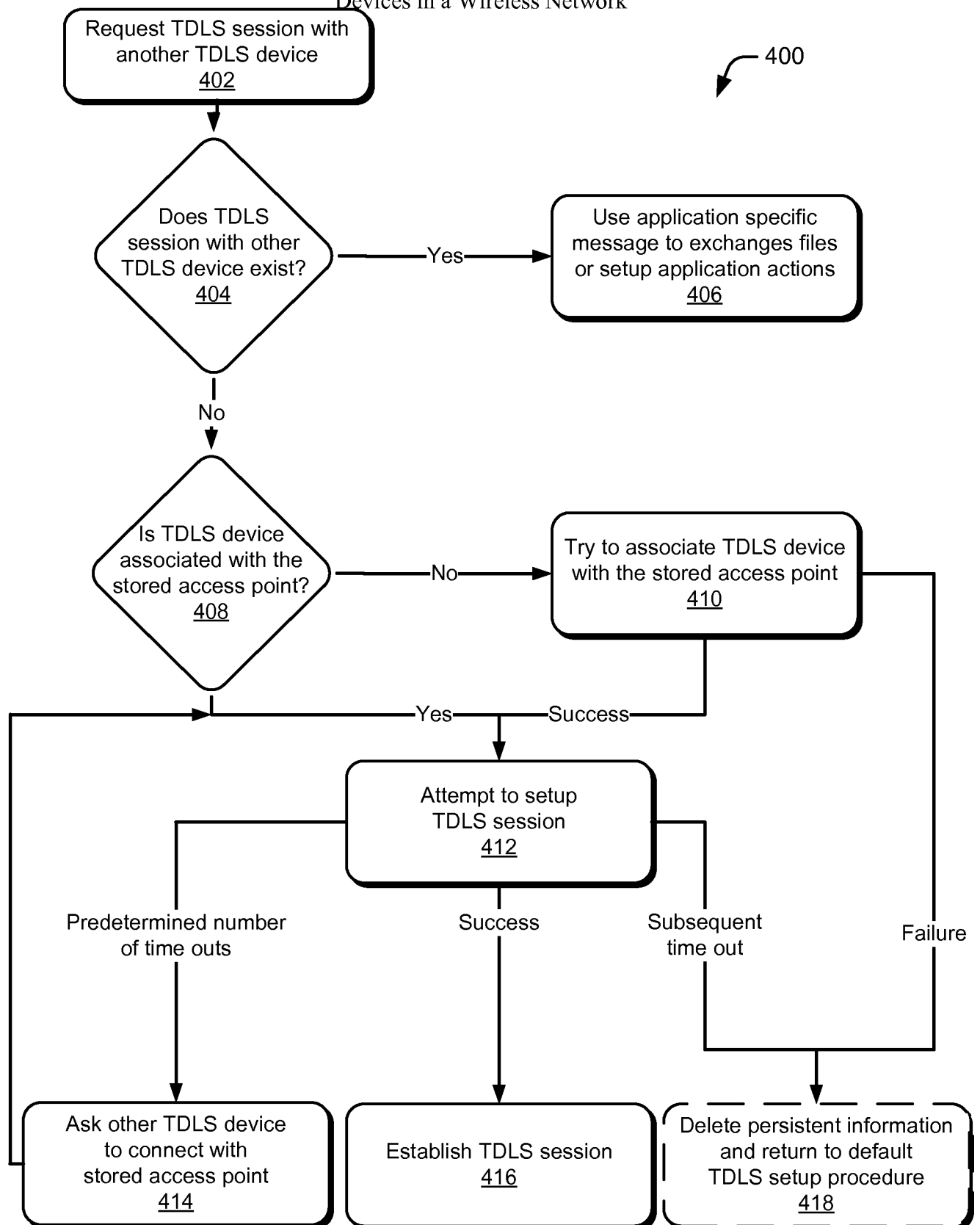


Fig. 4

Methods and Apparatus for Establishing a Tunneled Direct Link Setup (TDLS) Session Between Devices in a Wireless Network

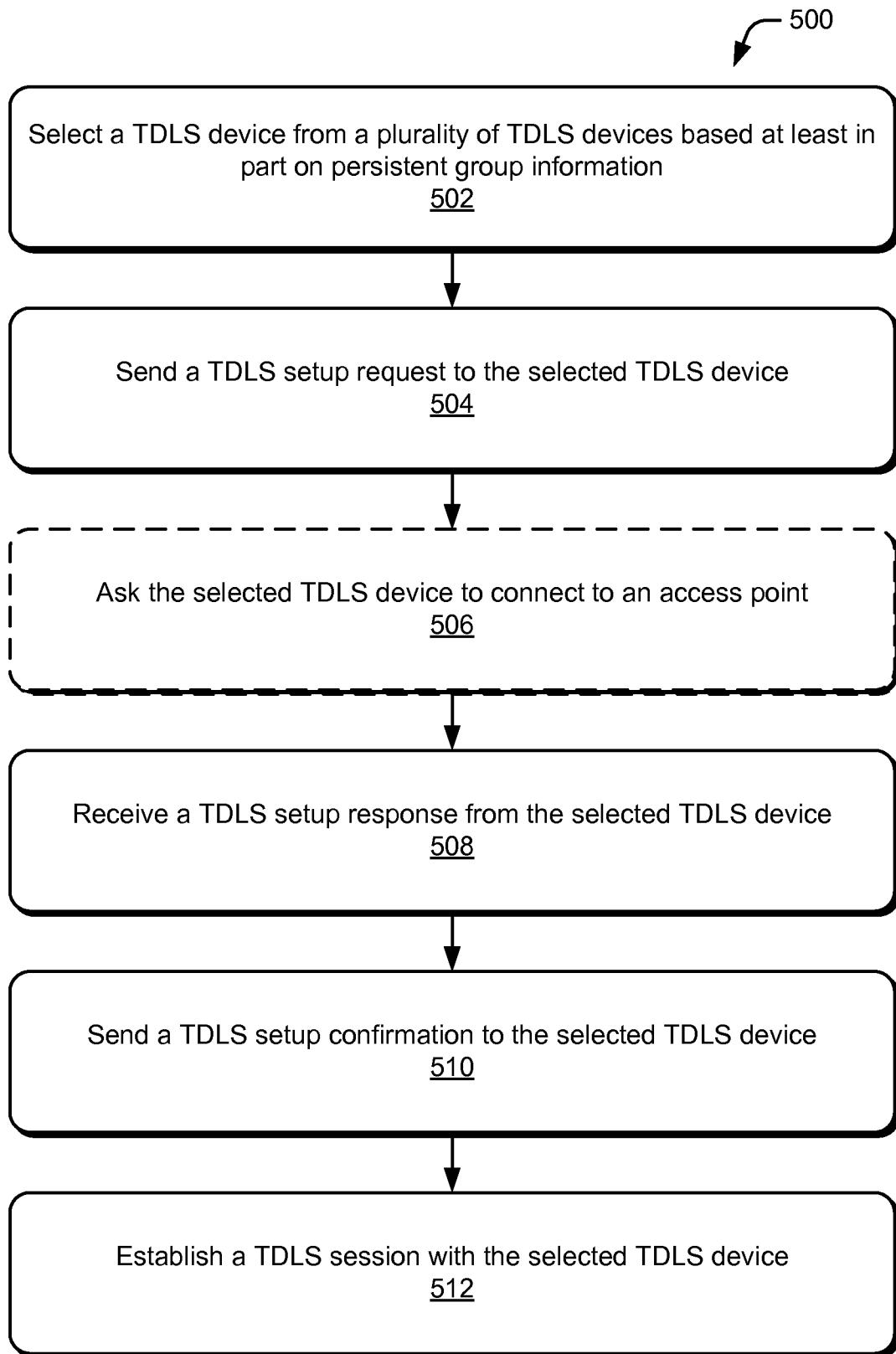


Fig. 5

Methods and Apparatus for Establishing a Tunneled Direct Link Setup (TDLS) Session Between Devices in a Wireless Network

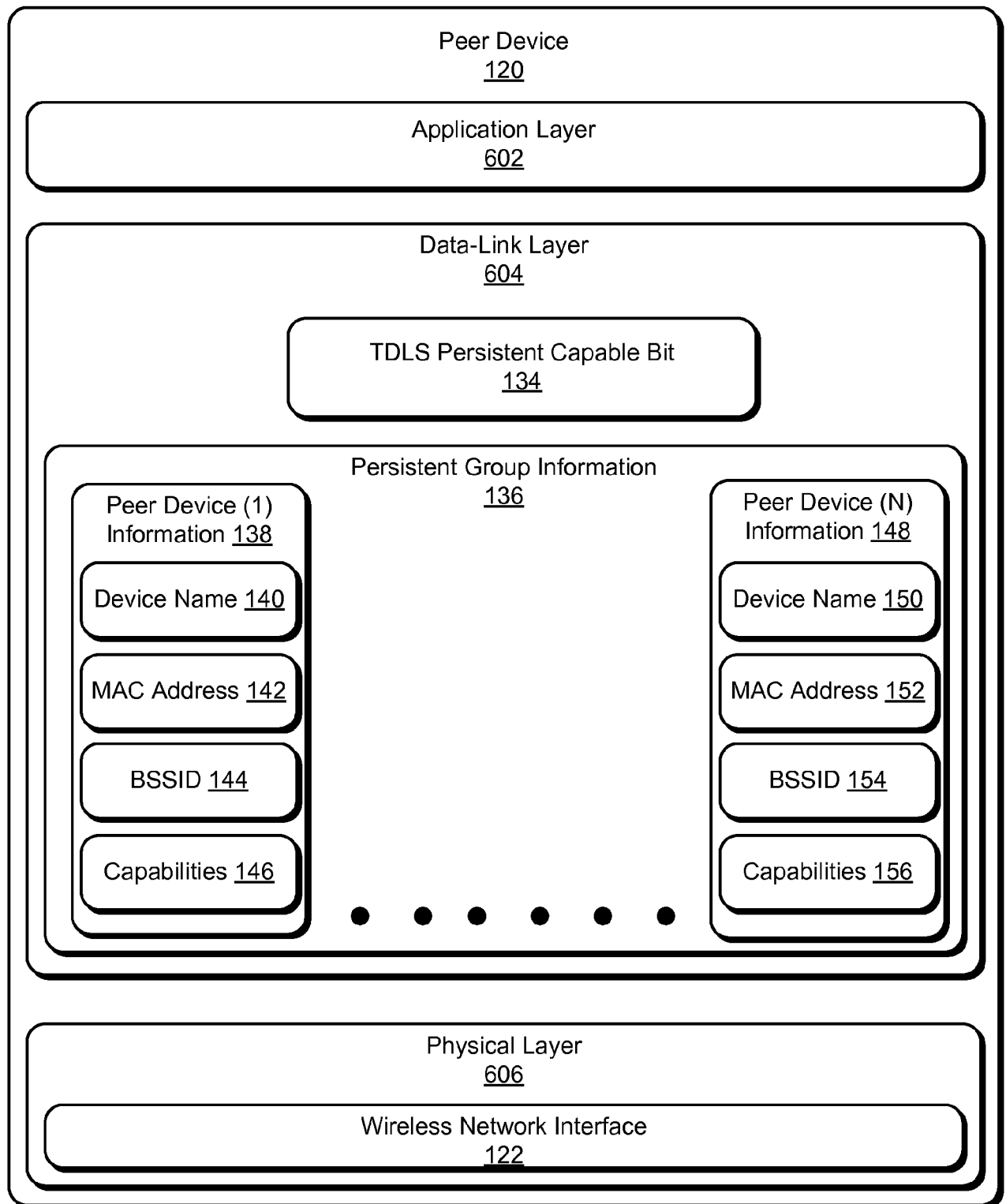


Fig. 6

Methods and Apparatus for Establishing a Tunneled Direct Link Setup (TDLS) Session Between Devices in a Wireless Network

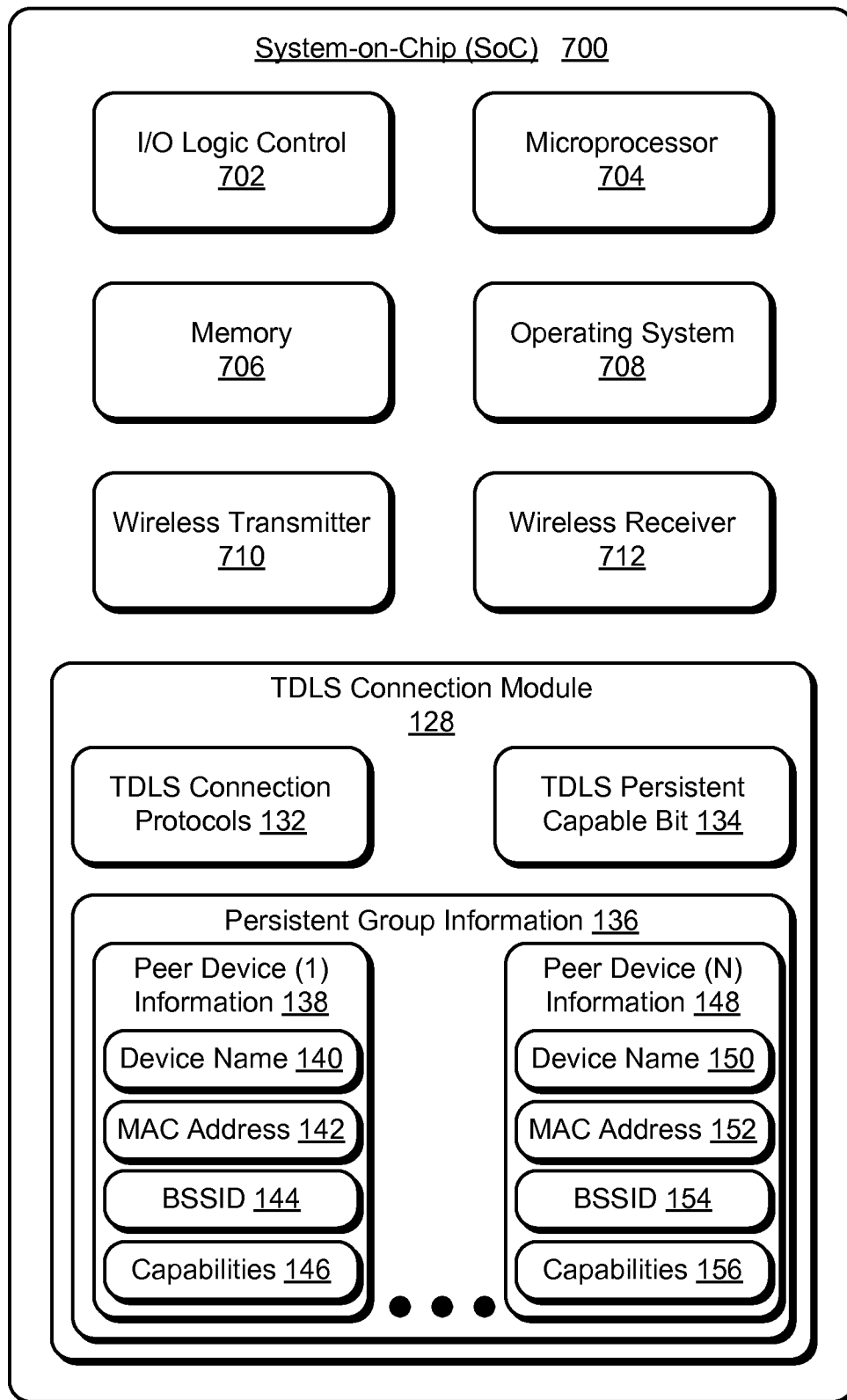


Fig. 7