

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
3 August 2006 (03.08.2006)

PCT

(10) International Publication Number
WO 2006/081215 A2

(51) International Patent Classification:
G06F 15/173 (2006.01)

(74) Agents: COCHRAN, David, B. et al.; Jones Day, North Point, 901 Lakeside Avenue, Cleveland, Ohio 44114 (US).

(21) International Application Number:
PCT/US2006/002367

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

(22) International Filing Date: 24 January 2006 (24.01.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/646,687 24 January 2005 (24.01.2005) US

(71) Applicant (for all designated States except US): DAIN-TREE NETWORKS PTY.LTD. [AU/AU]; 1 Dalmore Drive, Carribean Park, Socresby, VIC 3179 (AU).

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

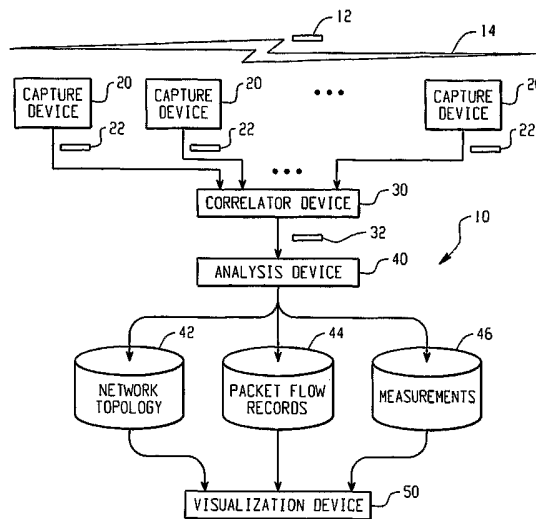
(72) Inventors; and

(75) Inventors/Applicants (for US only): CHOONG, Jason, Yew, Choo [AU/US]; 1926 Vinedale Square, San Jose, California 95132 (US). BARRON, Gregory, Joseph [AU/AU]; 42 Smith Street, Thornbury, VIC 3071 (AU). VAN GERREVINK, Dean [AU/AU]; 24 Rix Street, Glen Iris, VIC 3146 (AU). COBB, Peter, Graeme [AU/AU]; 24 Mandowie Crescent, Croydon, VIC 3136 (AU). WOOD, William, Raymond [AU/AU]; 20 Mason Street, South Yarra, VIC 3141 (AU). MORTON, Marcel [AU/AU]; 8 Wattleree Road, Ferntree Gully, VIC 3156 (AU).

Published:
— without international search report and to be republished upon receipt of that report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: NETWORK ANALYSIS SYSTEM AND METHOD



(57) Abstract: A packet-based wireless network visualization system includes a correlator subsystem, an analysis subsystem, and a visualization subsystem. The correlator subsystem generates correlated packet records representative of the order in which packets were transmitted in the wireless network. The analysis subsystem generates network topology data representative of the network topology, packet flow data representative of the flow of packets between devices at a MAC layer and across the network at the network layer, and measurement data relating to the packet flow data. The visualization subsystem generates a visual representation of the network topology, packet flows, and measurements.

WO 2006/081215 A2

NETWORK ANALYSIS SYSTEM AND METHOD

BACKGROUND AND SUMMARY

This disclosure generally relates to network analysis systems, and in particular relates
5 to wireless mesh network analysis systems and methods.

Wireless mesh networking is an emerging technology for enabling wireless
interconnections between a variety of devices, such as sensors, actuators, switches,
communication devices, and other network devices. The network devices may be
implemented to support a variety of applications, such as home automation, building and
10 industrial automation, environmental monitoring, data communication, etc.

Wireless mesh networks typically implement a dynamic topology in which devices
are associated and disassociated with other devices in the network and in which the roles and
responsibilities of such devices may change over time. Thus there are unique challenges to
developing, deploying and managing wireless networks.

15 Disclosed herein is a novel network analysis system and method that facilitates the
capturing of network data for analysis, measurement, and visualization. The capturing of
network data may be implemented in-band or out-of-band from the network. Captured
network data may be analyzed in real time or stored for later analysis. Measurements
obtained from the analysis may relate to network performance, device performance, route
20 performance, or other network characteristics. The topology of the network may be
graphically displayed, and measurement data may be accessible via the graphical display of
the network topology.

In one embodiment, a system for analyzing a packet-based network includes a
correlator processor that is configured to receive packet records corresponding to packets
25 communicated over a network and store the packet records in a data store. The correlator
processor is also configured to generate correlated packet records from the packet records
stored in the data store, the correlated packet records representative of the order in which the
packets were transmitted in the network.

In another embodiment, a system for analyzing a packet-based network includes a
30 wireless network analysis processing device that is configured to receive correlated packet
records representative of the order in which corresponding packets are transmitted in a
wireless network. The correlated packet records include media access control (MAC) layer
data and network layer data for each corresponding packet. The MAC layer data and network
layer data are processed to generate network topology data representative of the network
35 topology, generate packet flow data representative of the flow of packets between devices at

the MAC layer and across the network at the network layer, and measurement data relating to the packet flow data.

In another embodiment, a system for analyzing a packet-based network includes a packet-based wireless network visualization system. The packet-based wireless network visualization system includes a data store, an input/output subsystem including a display device, and a processing subsystem. The system is configured to receive network topology data, packet flow record data, and measurements data over the input/output subsystem and store the network topology data, packet flow record data, and measurements data in the data store. Based on the stored data, the system generates a visual representation of a network topology on the display based on the network topology data, generates a visual representation of packet flows within the network topology based on the packet flow records, and selectively displays measurement data related to the packet flows and network topology based on the measurements data. The visual representation of the network topology includes device objects, associations of device objects and a plurality of layer representations.

In another embodiment, a system for analyzing a packet-based network includes a plurality of capture devices configured to monitor packets communicated over the network and create the corresponding packet records. Each capture device includes a capture clock and each capture device is further configured to include a timestamp in each packet record corresponding to the capture clock time the capture device detects the start of frame of a packet. In one embodiment, the capture devices may communicate with a correlator processor in-band over the wireless network being observed. In another embodiment, the capture devices may communicate with a correlator processor out-of-band over another data network, e.g. a local area network (LAN). In another embodiment, the capture devices may simultaneously perform an active network device role and perform a passive sniffing role to capture network data.

DRAWINGS

Fig. 1 is a block diagram of an example network analysis system.

Fig. 2 is a block diagram of an example capture device.

Fig. 3 is a block diagram of another embodiment of a network analysis system.

Fig. 4 is an illustration of a multi-point correlation and filtering of network communication data.

Fig. 5 is an illustration of an example analysis device data structure.

Figs. 6-8 are example flow diagrams of network analysis device processes as used with the example analysis device data structure of Fig. 5.

Fig. 9 is an example visual representation of devices according to a network topology and corresponding packet flows.

Fig. 10 is an example visual representation of devices according to a network topology and corresponding application endpoint bindings.

5 Fig. 11 is a block diagram of another example network analysis system.

Fig. 12 is a flow diagram of an example process of creating packet records.

Fig. 13 is a flow diagram of an example process of creating correlated packet records.

Fig. 14A is a flow diagram of an example process of filtering packet records of packet records corresponding to duplicate packets and retransmitted packets.

10 Fig. 14B is a flow diagram of an example process of filtering packet records of packet records corresponding to duplicate packets and duplicate retransmitted packets.

Fig. 15 is a flow diagram of an example process of timestamp synchronizing packet records.

15 Fig. 16 is a flow diagram of an example process of generating packet stream data and packet flow data.

Fig. 17 is a flow diagram of an example process of detecting packet flow for a stream.

Fig. 18 is a flow diagram of another example process of detecting packet flow for a stream.

20 Fig. 19 is a flow diagram of an example process of determining a network topology and packet flow in the network topology.

Fig. 20 is a flow diagram of an example process of updating a network topology.

Fig. 21A is a flow diagram of an example process of generating a visual representation of a network.

25 Fig. 21B is a flow diagram of another example process of generating a visual representation of a network.

Fig. 22 is a flow diagram of an example process of generating a visual representation of device application bindings, endpoints and functions.

Fig. 23 is a flow diagram of an example process of generating personal area network (PAN) objects.

30 Fig. 24 is an example visual representation of PAN objects.

Fig. 25 is block diagram of an example operating model of a network analysis system.

Fig. 26 is an example breakpoint dialog window.

Fig. 27 is an example profile editor window.

Fig. 28 is an example profile properties window.

35 Fig. 29 is an example cluster properties window.

Fig. 30 is an example application layer decode window.

Fig. 31 is an example application layer measurements window.

Fig. 32 is an example visual representation of an application layer in a network.

Fig. 33 is an example device tree window.

5 Fig. 34 is an example device tree window with de-associations and re-associations indicia.

Fig. 35 is an example measurements window for providing information on the states of devices, streams and routes.

Fig. 36 is an example device context menu.

10 Fig. 37 is an example stream context menu.

Fig. 38 is an example route context menu.

Fig. 39 is an example APS binding context menu.

Fig. 40 is an example APS cluster context menu.

Fig. 41 is an example Expand context menu.

15 Fig. 42 is an example visual representation of a device tree.

Fig. 43 is an example visual representation of device re-associations.

Fig. 44 is a flow diagram of an example process of generating visual re-association indicia of device associations.

Fig. 45 is an example visual representation of routes.

20 Fig. 46 is an example visual representation of a selected device and corresponding routes.

Fig. 47 is an example visual representation of route selection.

Fig. 48 is an example visual representation of direct addressing in an APS layer.

Fig. 49 is an example visual representation of indirect addressing in an APS layer.

25 Fig. 50 is a visual device layout window.

Fig. 51 is a visual device layout window showing associations, routes, endpoints and bindings.

Fig. 52 is an example visual device layout window showing a network topology in three dimensions.

30 Fig. 53 is an example visual device layout window showing a network topology having endpoints that are visually represented as intrinsic elements of network device objects.

DETAILED DESCRIPTION

35 Fig. 1 is a block diagram of an example network analysis system 10. The network analysis system 10 includes one or more capture devices 20, a correlator device 30, an

analysis device 40, and a visualization device 50. The capture devices 20 capture packets 12 communicated over a wireless link 14 in a wireless network. The packets 12 are processed into packet records 22 that are provided to the correlator 30. The correlator 30, in turn, processes the packet records 22 and generates correlated packet records 32 that are provided to the analysis device 40. The analysis device 40 processes the stream of correlated packet records 32 to create network topology data 32, packet flow records 44 and measurements data 46. The network topology data 32, packet flow records 44 and measurements data 46 are accessed by a visualization device 50 to create a visual representation of the network and facilitate network analysis.

10 The correlator device 30, analysis device 40, and visualization device 50 may be implemented by executing software instructions on a single computer or on one or more computers in data communication over a network. Alternatively, specially designed hardware and/or software may be used to implement one or more of the correlator device 30, analysis device 40, and visualization device 50. Other hardware and/or software
15 implementations may also be used.

Network device transmissions over the wireless link 14 are bounded by a "personal operating space," i.e., a device range. Network device transmissions from a given device will be detected by other network devices operating within that personal operating space. Similarly, a given network device will detect transmissions from other network devices within its own operating space. In one embodiment, the personal operating space of each network device in the network intersects with the personal operating space of at least one capture device 20. In one embodiment, the capture devices 20 are dedicated passive packet sniffer devices. In a different embodiment, the capture devices 20 may be integrated into active network devices and simultaneously perform an active network device role and a passive sniffing role to capture network data. The simultaneous role of the capture devices 20 provides for active interrogation of the network by the sensor analysis system 10 to facilitate network analysis.

In one embodiment, the network analysis system 10 of Fig. 1 may be used to passively monitor the flow of packets 12 through the wireless network. The packet records 22 may include a complete packet 12, or may only include a subset of a complete packet, e.g., header information, media access control (MAC) information, and network information.

The statistical nature of radio transmissions may cause a single capture device 20 to miss an occasional packet transmission. When more than one capture device 20 can detect a particular packet 12 transmitted from a network device, the probability of missing a packet transmission is reduced. The overlapping of personal operation spaces of the capture devices
35

22, however, may result in duplicate packet detections when a packet is detected at multiple capture devices 20. Thus the correlator 30 receives the packet records 22 and filters the packet records 22 of packet records 22 corresponding to duplicate packets 12 and/or retransmitted packets 12. Packet records 22 corresponding to duplicate packets 12, i.e., a packet that is detected at multiple capture devices 20, may be used to calibrate timestamp information from each capture device 20.

In one embodiment, the calibrated timestamps are used to generate the correlated packet records 32 that comprise the packet records 22 placed in the order that their corresponding packets 12 were detected by the capture devices 20. Accordingly, the correlated packet records 32 are representative of the order in which the packets were transmitted in the network. In another embodiment, the correlated packet records 32 need not be placed in the order that their corresponding packets 12 were detected by the capture devices 20; instead, the representative of the order in which the packets 12 were transmitted in the network may be recreated by indexing the timestamp of each packet record 22.

The analysis device 40 receives the correlate packet records 32 and processes the records 32 to generate the network topology data 32, packet flow records 44 and measurements data 46, which is, in turn, used to detect the topology of the wireless network and to identify packet paths and the nature thereof through the wireless network. Identified packet routes are used to generate statistics on the flow of packets through the network.

The visualization device 50 may generate a visual representation of the network. The discovered routes and streams may be visually represented on the graphical representation of the network topology. A stream comprises all packets transmitted at the network layer between two devices. A route comprises a unique sequence of nodes traversed by a given stream of packets. Hence a stream will be comprised of one or more routes. The packet routes may be shown as traversing devices in the wireless network, and may be classified as successful or failed routes to aid troubleshooting, network planning and installation.

Fig. 2 is a block diagram of an example capture device 20. Each capture device 20 includes a capture clock 24 that is used to generate timestamps. The capture clock 24 may comprise a system clock within the capture device 20. A timestamp is a numerical representation of the time that the start of frame of an incoming packet 12 transmitted by a network device 16 was detected by the capture device 20. As a frame radio signal is received by the capture device 20, it is converted into a binary bit stream and temporarily stored in a data store, such as a memory. As the capture device 20 detects the start of frame for an incoming packet 12, a timestamp is also generated and placed in the data store. When the frame has been completely received, a time-stamped packet record 22 is created. The

packet record 22 may include the entire received packet 12 or a summary containing the particular packet 12 fields, such as MAC addresses and network address. In one embodiment, the packet record 22 includes the packet 12 and a timestamp 26.

In one embodiment, the capture clock 24 is a free-running independent clock and timestamp synchronization of packet records 22 is implemented in the correlator 30. In another embodiment, the capture devices 20 may communicate synchronization data to synchronize the capture clocks 24 in each capture device. In this embodiment, one capture device 20 may be designated a master capture device 20 and all other capture devices may calculate offsets or receive offsets to be added to their respective capture clocks 24 so that all capture clock 24 values are synchronized. The timestamp synchronization function may then be omitted from the correlator 30.

The timestamped packet records 22 are forwarded from the capture devices 20 to the correlator device 30. The capture devices 20 and the correlator device 30 may communicate in-band over the wireless network 14 or out-of-band over a separate network 18. An out-of-band network can be used to avoid traffic congestion that will modify the behavior of the network.

Fig. 3 is a block diagram of another embodiment of a network analysis system 10. In this embodiment, a host computer 60 implements the functionality of the correlator device 30, analysis device 40, and visualization device 50. The capture devices 20 communication with the host computer 60 via an out-of-band network 18, such as an Ethernet network.

Fig. 4 is an illustration of a multi-point correlation and filtering of network communication data. Due to the overlapping of personal operating spaces of the capture devices 20, a packet 12 transmitted by a single network device 16 will be detected by multiple capture devices 20 and thus multiple corresponding packet records 22 may be transmitted to the correlator device 30 for the same packet 12. For example, in Fig. 4, capture devices 20a - 20e selectively detect seven packets 12 transmitted during different time periods $t_a - t_g$. Thus duplicate packet records 22a - 22g are generated during time periods $t_a - t_g$, i.e., packet records 22a comprise three duplicate packet records; packet records 22b comprise four duplicate packet records, etc. The correlator device 30 functions to detect the duplicate packet records 22a - 22g and discards all but one of each duplicate packet record 22a-22g to construct correlated packet records 32a-32g representing packet transmissions by devices 16 in the network.

In one embodiment, the correlator device 30 also processes retransmitted packets by identifying and discarding retransmitted packets. Often transmitted packets may require acknowledgment by a receiving device, such as in the case in which a transmitting device

inserts a flag in those transmitted packets that requires acknowledgment. Unacknowledged packets may be retransmitted a certain number of times or until an acknowledgment signal is received by the transmitting device.

5 To generate the correlated packet records 32, the correlator device 30 stores each packet record 22 received from each capture device 20 in a data store, such as a cache. When a new packet record 22 is received, it is checked to determine whether it has already been received from the same capture device 20 by comparing the received packet record 22 to packet records 22 stored in the data store. If it has already been received from the same capture device 20, the original received packet record 22 is marked as being subject to retransmission and the received packet record 22 is discarded. Packet records 22 may be compared by comparing the full contents of the packet record 22, or by comparing a subset of the packet record 22. The subset may be used to generate 16 or 32-bit cyclic redundancy checksum of the entire packet record 22, for example. Other comparison techniques may also be used.

15 If the packet record 22 does not correspond to a retransmitted packet 12, the packet record 22 is compared to packet records 22 stored in the data store to determine whether the packet record 22 has been received from another capture device 20. The same comparison techniques described above may be used in this comparison. If the packet record 22 has been received from another capture device 20, then the received packet record 22 is discarded, and the original received packet record 22 is designated as a duplicate packet record.

20 If a packet record 22 does not correspond to a duplicate packet 12 or a retransmitted packet 12, then the packet record 22 is stored in the data store as a new packet record 22.

Before duplicate packet records 22 are discarded, they are used to perform timestamp synchronization. The correlation device 30 first designates one capture device 20 as master timing capture device 20 for the purpose of synchronization. All other capture records 22 from other capture devices 20 are time synchronized to master timing capture device 20 as packet records 22 are received. As duplicated packet records 22 are received, a timing offset is calculated between the time the packet record 22 was received from the master timing capture device 20 and the unsynchronized capture device 20. An offset is calculated and stored for each capture device 20 and applied to all subsequent packet records 22 received from each capture device 20, respectively.

30 Once a capture device 20 has become synchronized, it may then be used to synchronize other capture devices 20. This facilitates packet records 22 from capture devices 20 that do not detect duplicate packets with the master timing capture device 20 to

nevertheless become synchronized to the master timing capture device 20. The timing offsets may be adjusted with every duplicate packet record 22 detected.

Timestamp synchronization is not performed on packet records 22 that are classified as retransmissions. This avoids timestamp synchronization between an original packet record
5 received 22 from a capture device 20 and a packet record 22 corresponding to a later retransmission of the packet 12 detected at another capture device 20.

The correlator device 30 uses the timing offsets to normalize the timestamps for all packet records 22 in the outgoing correlated packet records 32. The potential exists for packet records 22 to be received from multiple capture devices 20 out of the order from
10 which they were actually transmitted. The normalized timestamps enable the correlator device 30 to reorder the packet records 22 in the outgoing packet stream to ensure they are forwarded to the analysis device 40 in the actual order they were transmitted. Reordering can be accomplished by storing packet records 22 intended for the outgoing correlated packet records 32 in a data store, such as a cache, for a time period before forwarding them to the
15 analysis device 40. This time period may be selected based on the longest time possible for a packet 12 to be transmitted by a network device 16, detected by a capture device 20 and forwarded to the correlator device 30 for analysis. The time period is therefore dependent on the actual network under test and the specific implementation of the analysis system 10 and thus may be determined empirically.

The correlator device 30 thus produces correlated packet records 32 that comprise an
20 ordered, time-stamped list of packet records 22 as detected across the wireless network, and free of packet records 22 corresponding to duplicate packets 12 or retransmissions of packets 12. The packet records 22 will appear in the ordered list, collected from different capture devices 20 in the network, in the order that their corresponding packets 12 were transmitted.
25 Thus the correlator device 30 ensures that for a network layer packet 12 that traverses across multiple network devices 16, each hop of its path will be represented in the correct order in the correlated packet list 32. For example, for a packet sent from network device A to network device D and traversing hops from network device A to network device B to network device C to network device D, the correlated packet records 32 will comprise three
30 packet records 22, one for each of the MAC layer segments from network devices A to B, B to C and C to D. The correlated packet records 32 will also be filtered of any packet records 22 corresponding to duplicate packets 12 or retransmitted packets 12.

Fig. 5 is an illustration of an example analysis device data structure 60, and Figs. 6-8 are example flow diagrams of network analysis device processes as used with the example
35 analysis device data structure 60 of Fig. 5. The example analysis device data structure 60 of

Fig. 5 and flow diagrams of Figs. 6-8 are described with reference to an example embodiment that is utilized in an IEEE 802.15.4 and ZigBee compatible wireless network. In other embodiments, other wireless networks may be analyzed.

The analysis device data structure 60 defines static object instances 62 and dynamic object instances 64. The data structure 60 further defines network device information 66 and measurement information 68.

The network device information 66 comprises a hierarchical NetworkTopology class 70, Structure class 74, Device class 76, Endpoint class 78, Stream class 80, Route class 82 and EP Binding class 84 hierarchically arranged as shown in Fig. 5. An object instance TableOfDevices 74 is associated with the NetworkTopology class 70. The measurement information 68 comprises a hierarchical MeasBase class 90, MeasSystem class 92, Meas class 94, MeasStruct class 96, ShowMeas class 98, DeviceMeas class 110, StreamMeas class 112, RouteMeas class 114 and EPMeas class 116 hierarchically arranged as shown in Fig. 5. An object instance theMeasSystem 100 is associated with the MeasSystem class 92. Likewise, object instances measStructDevice 102, measStructStream 104 and measStructRoute 106 are associated with the MeasStruct class 96, and the object instance theShowMeas 108 is associated with the ShowMeas class 98.

The packet capture subsystem block 122 of Fig. 6 generates the correlated packet records 32. The packet capture subsystem block 122 of Fig. 6 may be representative of the capture devices 20 and the correlator device 30. The analysis device 40 receives correlated packet records 32 from the correlator device 30. Each correlated packet record 32 is decoded to detect MAC, network and application layer information, as shown in step 124. The MAC layer information is used to detect changes to the network topology, and in particular detect new devices and/or associations between devices, as shown in step 126. The network layer information is used to detect the flow of packets through the wireless network and to create packet flow records. The decoded correlated packet records and packet flow records are then used in providing updates in the measurements data structures. Table 1 below provides an example list of decoded packet data.

Data	Description
Index	Index (sequence) number of the packet. The index is initialized to 1 at the beginning of a capture session. Indices are assigned sequentially to all packets received, and thus an index missing in the Packet List is indicative of a filter being applied.
Time	Time at which the packet was received, as provided by the capture device.

	The format is 'seconds.microseconds' where 'seconds' is the time in seconds since midnight, January 1, 1970, and 'microseconds' is an offset in microseconds from this time.
Src PAN	The MAC Source PAN field.
Src	The MAC source address field.
Dest PAN	The MAC Destination PAN field.
Dest	The MAC destination address field.
MAC Seq No	The MAC Sequence Number.
NWK Src	The NWK source address field.
NWK Dest	The NWK destination address field.
NWK Seq No	The NWK Sequence Number.
APS Src EP	The APS Source Endpoint.
APS Dest EP	The APS Destination Endpoint.
APS Profile	The APS Profile ID.
APS Cluster	The APS Cluster ID.
AF Seq No	Application Frame Sequence number.
Protocol	The appropriate protocol layer corresponding to the packet.
Packet Type	The actual packet type. For example, ZigBee NWK layer packet types may be "Command" or "Data."

Table 1: Example Decoded Packet Data

Steps 128, 130 and 132 provide an example implementation of step 126. Step 128 detects new devices and updates existing device information based on the MAC, network and application layer information. Step 130 detects changes to the network topology based on the MAC, network and application layer information. Step 132 detects new streams and endpoints based on the MAC, network and application layer information.

In one example embodiment, the analysis device 40 detects nodes that are added to the network by detecting ASSOCIATION frames in which a new network device searches for associations. The first existing network device that receives and responds to the ASSOCIATION frames from the new network device becomes the parent of the new device and a new device association is formed. The analysis device 40 detects this exchange of packets and adds the new device and association to the NetworkTopology 70 data structure. The source and destination network layer addresses are used to identify a packet stream that represents the flow of packets between two devices in the network. The measurements data

structure 96 maintains a list of all packet streams emanating from each device. When a new stream is detected, a new stream object 104 is added to the measurements data structure 96.

Step 134 extracts network object information. Step 134 typically involves a protocol-aware process, e.g., a process specific to a ZigBee network or 802.11 network, for example.

5 Steps 136, 138, 140 and 142 provide an example implementation of step 134 for a ZigBee network. Step 136 determines MAC duplicate suppressions and counts based on packet data. Step 138 obtains network layer measurements. Step 140 executes route and application support layer detection algorithms and measurement algorithms, examples of which are provided in Fig. 7 below. Step 142 detects packet routes in the network based on command
10 data.

In the example embodiment, the analysis device 40 collects MAC layer measurements for each device in the network. These measurements may include packet counts and received signal strength. For each stream, the analysis device 40 performs route detection to detect the routes taken by the flow of packets through the network, e.g., a sequence of hops taken by
15 one or more packets. Fig. 7, below, provides an example route detection process. The path taken by a particular packet is captured in a packet flow record. Once all hops for a given packet flow have been detected, a packet flow record is complete and is added to a packet flow list.

The measurements data structure 96 is updated when a complete packet flow record is
20 created. Each stream object 80 has one or more corresponding route objects 82 to represent the given path taken through the network. A complete packet flow record is therefore used to create a new route object 82 if one does not exist for the given path; otherwise, the packet flow record is used to update the measurements 114 for the corresponding route. For each route object 82, measurements may include statistics for packet counts and latency. The
25 statistics are updated for each complete packet flow record. In one embodiment, the analysis device 40 utilizes a timer to indicate measurement intervals for accumulating, latching and reporting the measurements that are collected.

Fig. 7 is a flow diagram 150 of example route and application support layer detection algorithms and measurement algorithms. For each unique network sequence number, step
30 152 tracks packets transitioning through the network across multiple packets. Step 154 classifies a route as failed if a packet does not arrive at a destination within a time period. The time period may be determined by the particular parameters of a given network under test. Step 156 classifies a route as malformed if a route is determined to have missing hops. Step 158 classifies a route as successful if a packet arrives at a destination with all sequential
35 hops. Step 162 performs network layer duplicate detection and suppression.

In the embodiment, the analysis device 40 determines the source and destination network layer addressing. Based on the source and destination addresses the analysis device 40 identifies a matching stream object 80. If a corresponding stream object 80 does not exist, one is created.

5 The analysis device 40 compares MAC addresses and network addresses. If the source MAC address matches the source network address then the packet record corresponds to the first hop of a new packet flow and a new packet flow record is created, updated with information on the first hop and added to the list of temporary packet flow records.

10 If the destination MAC address matches the destination network address then the correlated packet record 32 corresponds to the last hop of a packet flow. The analysis device 40 identifies the packet flow record in the list of temporary packet flow records and transfers the packet flow record to the packet flow record list. This triggers a completion of the packet flow processing which causes the route measurements data structures 114 to update and triggers application layer measurement processing.

15 If the destination MAC address matches the destination network address more than once, then the analysis device 40 determines that duplicate network packets were transmitted and reports duplicate packet measurements.

20 If the correlated packet record 32 does not correspond to the first or last hop, then the analysis device 40 identifies the temporary packet flow record for correlated packet record 32 and updates the temporary packet flow with this latest intermediate hop.

25 In one embodiment, there is an aging process to process incomplete packet flow records at the end of every measurement interval. Any temporary packet flow record that has not been updated for a configurable timeout interval is marked as an incomplete route and transferred from the temporary list of packet flow records to the list of completed packet flow records.

30 Thereafter, step 160 may perform application layer processing on the identified routes. In the example embodiment, application layer functionality is represented as endpoints that embody the functionality of a particular network device, and a single network device may support multiple endpoints. The analysis device 40 processes valid application layer packets and ignores other packets. The application layer packets are analyzed to determine the endpoints identified by the packets. If a new endpoint is detected, a corresponding device object 76 in the NetworkTopology data structure 70 is updated to reflect the new endpoint. Application layer information, such as a supported application and attributes, about the endpoint is stored therein.

On completion of a network layer packet flow record, the application layer information relevant to the flow record is updated. Such information may include the source and destination endpoint. Flows between endpoints are stored as application endpoint (EP) Binding objects 84 as part of the stream object within 80 the overall measurements data structure 94.

Fig. 8 is a flow diagram 170 of the measurement subsystem and visualization subsystem processes and supporting data structure. A measurements subsystem 172, which may be implemented by software instructions embedded in the analysis system 40, comprises user-definable measurement controls 174 that provide a user with measurement controls and allow the user to configure measurements. Step 176 registers measurements at the end of a measurements interval, and step 178 presents the measurements to the user via a measurements environment, such as an application window. A visualization subsystem 180 provides a visual representation of the network to facilitate analysis. An example visual representation may include the network structure or topology, device objects, stream and route illustrations, and APS connectivity.

Fig. 9 is an example visual representation 190 of devices according to a network topology and corresponding packet flows. The results of the analysis device 40 processing can be revealed graphically using a device visualization diagram. Illustrated is a simple five node network, with node 192 defining the top of a device tree and having two children nodes 194 and 196. Node 196, in turn, has children nodes 198 and 200. The five nodes in a tree structure represent the device visualization structure.

Detected packet flows, data for which is stored in the packet flow records 44, may be mapped onto the visual representation 190. Each packet flow record 44 defines a packet path taken through the network. Each packet flow is depicted using a spline 202, 204 or 206 that connects through each of the devices traversed by the packet as it traveled through the network from source to destination. Multiple packet flow records can be shown simultaneously by drawing multiple splines, as shown in Fig. 9. Three routes are overlaid on the visual representation 190. Route 202 defines a route from device 194 to device 192 to device 196 to device 200. Route 204 defines a route from device 194 to device 196 to device 200. Route 206 defines a route from device 194 to device 198 to device 200.

In one embodiment, each spline is rendered with unique visual indicia, such as a different color. Additional information about a packet flow can be examined by selecting a spline 202, 204 or 206. Additional information can include highlighting each of the nodes traversed by the packet flow, a popup screen with statistics on the route such as the number of packet flows that were observed on the given route, or other measurements.

The baseline visualization represents the two-dimensional topology of the wireless mesh network as described in the network topology data 42 maintained by the analysis device 40. The topology may comprise a star, tree, mesh or combination thereof. Regardless of the structure, the network topology represents the devices in the network and the associations
5 between them.

In one embodiment, the visualization device 50 may be implemented to visualize a ZigBee network. The visual representation 190 provides the basis for visualizing the result of the network (NWK) layer analysis. The visual representation 190 depicts formal associations and may also facilitate the processing of network layer analysis.

10 Fig. 10 is an example visual representation 210 of devices 212, 214, 216, 218, and 220 according to a network topology and corresponding application endpoint bindings 222, 224, 226, 228, 230 and 232. The packet flow records may also identify additional information about the packet flows. For example, in wireless mesh networks, end devices often contain application endpoints that encapsulate a specific application or function. A first
15 device 218 may include a switch 226 and a second device 220 may include a light 230. The packet flow record analysis can reveal that a given packet flow corresponds to a binding between two application endpoints, such as a switch 226 to a light 230 between devices 218 and 220. Thus the visualization representation 210 may provide details of application endpoint bindings and be labeled with the detailed functions of each endpoint. This enhances the
20 usefulness of the network visualization by relating it to the function of the devices themselves.

Fig. 11 is a block diagram of another example network analysis system 10, similar to the example of Fig. 3. A plurality of capture devices 20 each include a wireless communication subsystem 240, a processing subsystem 242, a data store 244, and an out-of-
25 band communication subsystem 246. The wireless communication subsystem 240, processing subsystem 242, data store 244, and the out-of-band communication subsystem 246 are in data communication via a bus system 248. The bus system 248 may comprise parallel and serial data busses. The wireless communication subsystem 240 detects packets communicated over a wireless link via an antenna 250.

30 Packet records are communicated to a computer 60 via a network 18, such as an Ethernet network. The computer 60 comprises an out-of-band communication subsystem 260, a processor 262, a data store 264, an I/O subsystem 266, and a data bus 268. The I/O subsystem includes a user-input device 270, such as a keyboard, and a display 272. The out-of-band communication subsystem 260 may also be part of the I/O subsystem 266. The
35 computer system 60 may include software instructions stored in the data store 264 that upon

execution by the processor 262 provide the functionality of the correlator device 30, analysis device 40, and visualization device 50 as described above.

Fig. 12 is a flow diagram 280 of an example process of creating packet records. Step 282 monitors network traffic for transmitted packets, and step 284 determines if a packet has been detected. In one embodiment, a packet is detected upon receiving a start of frame. If a packet is not detected, the process returns to step 282. If a packet is detected, step 286 receives the packet, and step 288 creates a time-stamped packet record. Step 290 transmits the packet record to the correlator device, and the process then returns to step 282.

Fig. 13 is a flow diagram 300 of an example process of creating correlated packet records. Step 302 receives packet records corresponding to packets communicated over a network. Step 304 filters the packet records for packet records corresponding to duplicate packets and packet retransmissions. Step 306 generates correlated packet records from the filtered packet records generated in step 304.

Fig. 14A is a flow diagram 310 of an example process of filtering packet records of packet records corresponding to duplicate packets and retransmitted packets. Step 312 compares received packet records to stored packet records. The packet records may be stored in a data store, such as a cache memory. Step 314 determines if a received packet record corresponds to a retransmitted packet. If a received packet record corresponds to a retransmitted packet of a stored packet record, the received packet record is discarded and the stored packet record is classified as a retransmitted packet record in step 316.

If a received packet record does not correspond to a retransmitted packet of a stored packet record, then step 318 determines if a received packet record corresponds to a duplicate packet of a stored packet record. If a received packet record corresponds to a duplicate packet of a stored packet record, then step 320 timestamp synchronizes the received packet record and the stored packet record, discards the received packet record, and classifies the stored packet record as a duplicate packet record. If a received packet record does not correspond to a duplicate packet of a stored packet record, then the packet record is stored in step 322.

Fig. 14B is a flow diagram 311 of an example process of filtering packet records of packet records corresponding to duplicate packets and duplicate retransmitted packets. In this embodiment, packet records received from a particular capture device corresponding to retransmitted packets are stored for further analysis. Duplicate packet records corresponding to the retransmitted packets, e.g., packet records generated by other capture devices that correspond to the retransmitted packet, are discarded.

Step 313 compares a received packet record for a first capture device to stored packet records received from the first capture device. Step 315 determines if the received packet record corresponds to a retransmitted packet record. If so, then step 317 stores the received packet record and classifies the received packet record and stored packet record as retransmitted packet records.

If the received packet record does not correspond to a retransmitted packet record, then step 319 determines if the received packet record corresponds to a duplicate packet record. If the received packet record does not correspond to a duplicate packet record the packet record, then step 321 discards the packet record.

If the received packet record does correspond to a duplicate packet record the packet record, then step 323 determines if the received packet record corresponds to a retransmitted packet from other capture devices. If not, then step 325 timestamp synchronizes the received packet record and the stored packet record, discards the received packet record, and classifies the stored packet record as a duplicate packet record. If, however, the received packet record corresponds to a retransmitted packet from other capture devices, then step 327 discards the received packet record.

Fig. 15 is a flow diagram 330 of an example process of timestamp synchronizing packet records. Step 332 designates a first capture device as a master capture device. Step 334 calculates a timing offset representative of the time difference between which a corresponding packet was received by the master capture device and a second capture device. Step 336 applies the timing offset to all subsequent packets received by the second capture device. The flow diagram 330 of Fig. 15 may be further generalized to multiple capture devices, e.g., third, fourth and fifth capture devices, etc.

Fig. 16 is a flow diagram 340 of an example process of generating packet stream data and packet flow data. Step 342 receives correlated packet records. Step 344 processes MAC layer data and network layer data. Step 346 generates packet flow data representative of the flow of packets between devices at the MAC layer. Step 348 generates packet flow data representative of the flow of packets across the network at the network layer.

Fig. 17 is a flow diagram 350 of an example process of detecting packet flow for a stream. Step 352 identifies packet streams based on a source network layer address and a destination network layer address in a packet record. Step 354 compares a source MAC address and a destination MAC address to the source network layer address and the destination network layer address. Step 356 detects packet flows for each stream based on the comparison.

Fig. 18 is a flow diagram 360 of another example process of detecting packet flow for a stream. Step 362 compares a source MAC address and a destination MAC address to a source network layer address and a destination network layer address in a correlated packet record. Step 364 determines if the source addresses match. If the source addresses match, step 366 creates a new packet flow record and adds packet flow data corresponding to a first hop of the packet in the network. If the source addresses do not match, step 368 determines if the destination addresses match. If the destination addresses match, then step 370 adds packet flow data corresponding to a last hop of a packet in the network. If the destination addresses do not match, then step 372 adds packet flow data corresponding to an intermediate hop of a packet in the network.

Fig. 19 is a flow diagram 380 of an example process of determining a network topology and packet flow in the network topology. Step 382 receives correlated packet records. Step 384 determines network topology changes and identifies new devices based on the correlated packet records. Topology changes may be inferred implicitly based on the flow of packets through the network and addressing information stored in the packet, or may be detected based on explicit association requests and response messages as shown in Fig. 20 below. Step 386 extracts network object information from the correlated packet records. Step 388 generates packet flow data representative of the flow of packets between the objects.

Fig. 20 is a flow diagram 390 of an example process of updating a network topology. Step 392 identifies new devices in the network from correlated packet records related to association requests and responses. Step 394 associates new devices in the network topology according to the device association established by the association request.

Fig. 21A is a flow diagram 400 of an example process of generating a visual representation of a network. Step 402 generates a visual representation of a network topology. Step 404 generates a visual representation of device objects and associations of device objects. Step 406 generates a visual representation of layers. The layer may include a MAC layer, a network layer, and an application layer. Step 408 generates visual representation of packet flows within the network topology. Step 410 selectively displays measurement data.

Fig. 21B is a flow diagram 411 of an example process of generating a visual representation of a network. Step 413 generates a visual representation of device objects and associations. This step is illustratively directed to the MAC layer. Step 415 generates a visual representation of packet flows within the network topology. This step is illustratively directed to the network layer. Step 417 generates a visual representation of end-to-end

communications across the network. This step is illustratively directed to the application layer. Step 419 selectively displays measurement data.

Fig. 22 is a flow diagram 420 of an example process of generating a visual representation of device application bindings, endpoints and functions. Step 422 generates a visual representation of bindings between application endpoints. Step 424 generates a visual representation of device functions associated with the application endpoints.

Fig. 23 is a flow diagram 430 of an example process of generating personal area network (PAN) objects. Step 432 generates PAN objects indicating identified PANs within the network topology. Step 434 generates PAN fragment objects indicating network fragments. Step 436 selectively displays PAN objects and PAN fragment objects.

Fig. 24 is an example visual representation of PAN objects. Before devices can communicate in a wireless mesh network they must join a PAN. In an example ZigBee network, only devices on the same PAN can communicate. Multiple PANs, however, may share the same wireless channel, and any device receiving a packet from a different PAN will discard the packet. This allows multiple logical networks to share the same physical network, e.g., a channel or a specific frequency range.

The analysis device 50 detect packets flowing on multiple PANs simultaneously and the detected network topology is broken out at the top level into multiple PANS. PAN objects are used to display one PAN at a time. The user can select which PAN to display by right-clicking on the PAN icon, which displays a list of detected PANs to select from. When the user selects a different PAN the associated network topology is shown.

A PAN fragment object is used to show partially formed network fragments for the selected PAN. This is a partial part of the network where the analysis device 50 has not yet deduced how PAN fragment connects to the rest of the network. A PAN fragment object may be used to show a partial fragment of the network that has been observed by the analysis device 40 and for which the analysis device 40 does not currently have adequate information to determine where the corresponding PAN attached to the rest of the network. Such a condition may occur when the formation of the network was not observed by the network analysis system 10. For example, the network analysis system 10 may have failed to detect some of the ASSOCIATION requests and responses. An identified PAN fragment may be the result of part of the network being formed prior to data collection by the capture devices 20, or part of the network being out of range of any capture devices 20.

A first PAN object 442 corresponds to an identified PAN in the network. The corresponding PAN of the PAN object 442 comprises network devices 446 and 448 as

indicated by association lines 450 and 452. The second PAN object 444 is a PAN fragment object, indicating a network fragment. The corresponding PAN fragment of the PAN object 444 comprises network devices 454 and 456 as indicated by association lines 458 and 460. Network devices 446, 448, 454 and 456 communicate data via paths 462.

5 Figs. 25-53 describe another example embodiment of the network analysis system 10 and corresponding methods described above. The example embodiment of Figs. 25-53 describes an implementation for monitoring and analyzing ZigBee networks. Other embodiments, however, may also be used to monitor networks of different protocols.

10 Fig. 25 is a block diagram of an example operating model 500 of another example network analysis system 10. Fig. 25 illustrates components of the network analysis system 10 and its corresponding environment, and how data from live networks or previously captured files are processed by these different components. Data from a live network 502 may be monitored by capture hardware 504 as configured by channel source and channel selection controls 506. Collected data may be stored in a capture file 508. The capture file
15 508 may be configured according to logging controls 510. The capture file 508 may comprise actual packets 12, packet records 22 and/or correlated packet records 32. An example capture file format for a ZigBee network is provided in Table 2 below. Other capture file formats may also be used, depending on the network analysis system 10 features implemented and depending on the network communication protocol.

20

Field	Length	Type	Description
Sequence	1-10	32-bit unsigned decimal integer.	This is the timestamp of the packet. It increments as each packet is placed into the file.
Timestamp	1-10.1-6	32-bit unsigned decimal integer.32-bit unsigned decimal integer.	This is the number of the packet's arrival. The first integer is the time in seconds since midnight, January 1, 1970. The second integer is an offset in microseconds from this time. The integers are separated by a period ('.').
Length	3	8-bit unsigned decimal integer.	This is the number of octets represented in the <i>Data</i> field.
Data	2-250	Concatenation of characters.	This is the packet data. Its length is determined by the <i>Length</i> field. The last two octets will be masked to 0xffff.

LQI	5	16 bit unsigned decimal integer	This is the Link Quality Indicator (LQI) of the packet.
FCS	1	0 or 1	This is the Frame Check Sequence (FCS) correctness of the packet. If 1 the FCS was correct, if 0 the FCS was incorrect.

Table 2: Example Capture File Format

Playback and analysis of the capture file 508 may be controlled via capture and playback controls 512. The capture file 508 may be analyzed by a correlator device 30 and/or an analysis device 40 to determine a device tree 514 that describes the network topology. Measurement controls 516 are used to manage the collection and analysis of measurements 518 that are, in turn, used to generate visual representations, including a visual device tree 520 and a visual device layout 522.

Packet data may also be analyzed via a packet decoder and display filter 524. Based on the input to the packet decoder and display filter 524, a packet list window 526, a packet decode window 528 and a packet data window 530 may display the decoded packet information and data.

The network analysis system 10 provides for the playback of capture files 508 previously captured from live networks. In one embodiment, the network analysis system 10 has a live network analysis mode and a post-analysis mode. The live network analysis mode analyzes network data and facilitates network analysis in near real-time. The post-analysis mode facilitates post-analysis of capture files previously captured from a live network by either a user or a third party.

In one embodiment, breakpoints may be added during live capture or during playback. Fig. 26 is an example breakpoint dialog window 540. During playback of a capture file 508, a packet may be selected via selection menu 542 and a breakpoint may be added before or after the selected packet. A breakpoint description may be added via a text field 544. Breakpoints may be added automatically during live capture upon the occurrence of a user-defined event, such as the occurrence of a BACON frame or a device association.

A breakpoint comprises of a user-definable string associated with the given location in the capture file. An entry is added to the capture file to represent the breakpoint. During playback, whenever a breakpoint is detected in the capture file 508, playback pauses and the breakpoint text string is displayed. Breakpoints may also be ignored by selecting a breakpoint disable mode.

Fig. 27 is an example profile editor window 550. In one embodiment, the network analysis system 10 defines ZigBee application profiles. These profiles are used to dynamically decode and analyze application layer messages. The profile editor window 550 supports the editing of an application profile. In one embodiment, each application profile is stored in a separate file. The profile editor window of Fig. 27, for example, is depicted as displaying a Home Control Lighting application profile.

The profile editor comprises a profile definition and a cluster definition. Clusters indicate the set of messages supported by the application profile. For each profile there may be one or more clusters.

Fig. 28 is an example profile properties window 552. A Profile ID field 554 displays the identifier that is used in the ZigBee APS Header to identify the profile. A Name field 556 displays a full name of the profile. An Abbreviation field 558 displays an abbreviated name of the profile. A Description field 560 displays additional text to describe the profile. The fields may be edited at the option of the user.

Fig. 29 is an example cluster properties window 562. A Cluster ID field 564 displays an identifier that is used in the ZigBee APS Header to identify a cluster. A Name field 556 displays a name of the cluster. A Description field 568 displays additional text to describe the cluster. The fields may be edited at the option of the user.

Fig. 30 is an example application layer decode window 570. When the packet decoder 524 is decoding application layer packets it will use the profile files to look up the Profile ID and Cluster ID fields 554 and 564 so that the names defined in these files are shown rather than just the hex values of these fields, as illustrated by the highlighted text "Home Control Lighting" 572.

Fig. 31 is an example application layer measurements window 574. In the example embodiment of Figs. 25-53, measurements are collected on a per cluster basis, based on the clusters defined profile definition files. When an APS message is detected on a particular endpoint, the network analysis system will automatically look up the profile ID and cluster ID in the relevant profile definition file and obtain and maintain measurements, such as packet count, etc., on a per-cluster basis. The text string "Home Control Lighting" 576 indicates the profile.

Fig. 32 is an example visual representation 578 of an application layer in a network. As application layer messages are detected by the network analysis system 10, they may be displayed on the visual representation 578. These messages may be shown as application layer bindings between application endpoint pairs. When the user selects one of these

bindings the visualization 578 will display the profile names and cluster names based on the definitions stored profiles and as illustrated by text string 580.

Fig. 33 is an example device tree window 582. The device tree window 582 provides a tree-view of the network topology. Each node may display its short address field 584, long address field 586, and an information field 588. The example information field indicates whether a device is currently accepting associations. In the example embodiment of 25-53, the device tree window 582 provides a network and device-centric view of an 802.15.4 or ZigBee network. The displayed device tree is created from data collected from a live capture session or from data stored in the capture file 508, and processed by the analysis device 40. Each level of the device tree may be expanded or collapsed as implied by the tree display format.

Other information may also be displayed in the information field. Fig. 34 is an example device tree window 590 with de-associations and re-associations indicia 592 that indicate whether a device has re-associated or disassociated.

Fig. 35 is an example measurements window 600 for providing information on the states of devices, streams and routes. The measurements window 600 provides a multi-column display with the first column 602 containing an expandable/collapsible network tree hierarchy in which the network objects are listed in a device tree 604. The hierarchy supported in the tree is PAN, device, stream, route type, and route. Packets on a given route are tracked using the source device and the NWK layer sequence number. A plurality of measurements 606 may be displayed, including transmission measurements, basic packet measurements, route discovery measurements, packet performance measurements, latency measurements, physical measurements, packet measurements, and broadcast measurements. Other measurements that facilitate network analysis may also be collected and displayed.

Routes may be classified according to various types. Example route types include successful, malformed, malformed-s, reroute, and failed. A successful route classification occurs when a given packet correctly arrives at its destination and all hops from source to destination device are detected. A malformed route classification occurs when a given packet correctly arrives at its destination but not all hops are detected. For this route type, the first hop is detected. A malformed-s route classification occurs when a given packet correctly arrives at its destination but not all hops are detected, including the first hop. A reroute classification occurs when packets arrive at the destination via a re-route. A failed route classification occurs when a given network layer packet is detected at one or more hops but the packet is not detected arriving at its intended destination.

Each level of the measurements hierarchy can also be provided with a right-click context menu. Context menus enable the selection and highlighting of corresponding items in the visual device tree 604, or provide shortcut packet filter operations to filter the packets shown in the packet list based on the selected item in the measurements window 600.

5 Different context menu items are available for different items.

Fig. 36 is an example device context menu 610. The device context menu 610 is available by right-clicking on a device object 612 in the measurements window 600. The device context menu 610 includes MAC layer filters 614 to match all packets where the selected device is the Source, Destination, or either, and a MAC layer filter 616 to match
10 when this device has participated in the MAC layer association sequence. The device context menu 610 also includes NWK layer filters 618 to match all packets where the selected device is the source, destination, or either a source or destination.

Fig. 37 is an example stream context menu 620. The stream context menu 620 is available by right-clicking on a stream object 622 in the measurements window. The stream
15 context menu 620 includes NWK layer filters 624 to match all packets corresponding to the selected stream, i.e., all packets between the given source and destination, and a select visual 626 menu to highlight a given source and destination device in a visual device tree.

Fig. 38 is an example route context menu 630. The route context menu 630 is available by right-clicking on a route object 632 in the measurement window 600. The route
20 context menu 630 includes NWK layer filters 634 to match all packets corresponding to the associated stream and a select visual 636 menu to highlight the given source and destination device in a visual device tree and to select and highlight the corresponding route in a visual device tree.

Fig. 39 is an example APS binding context menu 640. An APS binding represents all
25 of the APS layer packets flowing between two APS endpoints on two different devices. The APS binding context menu 640 is available by right-clicking on an APS binding 642 in the measurements window 600. The APS binding context menus 640 include NWK layer filters 644 to match all packets corresponding to the associated stream, APS Layer Filters 646 to match packets flowing from a source endpoint and/or to a destination endpoint, and a select
30 visual 648 menu to select and highlight the corresponding APS binding in a visual device tree.

Fig. 40 is an example APS cluster context menu 650. An APS cluster identifies a specific class of application layer attributes being exchanged by two APS endpoints on two different devices. The APS cluster context menu is available by right-clicking on an APS
35 cluster object 652 in the measurements window 600. The APS cluster menu 650 includes an

APC layer filter 654 to match all packets on the given cluster between the given source device/endpoint and the given destination device/endpoint.

Fig. 41 is an example Expand context menu 660. The expand context menu 660 is available for the entire Measurements Window 600 and is not specific to different levels of the measurement hierarchy. The expand context menu 660 may be used to select to what level the measurements window 600 is expanded and collapsed and hence determines what is shown at any point in time. For example, an expand menu 664 may be used to expand or collapse the measurements window 600 to the device level, streams level, routes level, endpoints level, or to all levels.

Fig. 42 is an example visual representation of a device tree 700. The visual devices tree 700 provides a graphical rendition of network topology and information flows between devices 702, 704, 706 and 708 that define the PAN 710. Devices are added dynamically based on 802.15.4 association response messages. Lines 712, 714 and 716 are shown between parent and child devices to indicate a MAC layer association. Routes, such as route 720, may be displayed on the visual device tree 700. Route displays can be filtered to show only those routes to and from a given device by selecting the device on the visual device tree 700. A given route, such as route 720, can be selected by clicking on the route line so that each device traversed by the route is highlighted. Application endpoints may also be shown. For example, application endpoints 722, 724 and 726, associated with devices 702, 706 and 708, respectively, are illustrated. Addressing scheme may also be indicated, all illustrated by an indirect addressing object 730.

In one embodiment, a single PAN is displayed at any one time. The current PAN ID for the PAN object 710 is represented by a rectangle as the root of the tree. A selectable list of available PANs for display may be displayed by right-clicking on the PAN object 710. Selecting a PAN ID will display a visual device tree for the selected PAN.

Fig. 43 is an example visual representation 740 of device re-associations. In an 802.15.4 network it is not uncommon for a device to lose its association and re-associate with another device in the network. For re-associated devices, their previous location in the visual device tree 700 is indicated by visual re-association indicia, such as an offset color. An arrow linking the old location to the new location in the visual device tree 700 with an arrowhead indicating the old and new location may also be displayed. A chain of linked re-associated nodes may be used to display multiple re-associations of the same device. For example, in Fig. 43, a device 742 was previously a child of device 744, and was re-associated to device 746. The address of the device 742 was changed from 2511 to 155a, and thus upon re-association the device appears as device 748. Accordingly, the device objects 742, 744 and

748 are displayed with an offset color. The visual re-association indicia may be selectively displayed.

Fig. 44 is a flow diagram 750 of an example process of generating visual re-association indicia of device associations. Step 752 generates visual re-association indicia
5 indicating a current device association and a previous device association. Step 754 selectively disables display of the re-association indicia.

Other visual representations may also be generated. For example, if a particular device is retransmitting a packet, the corresponding device object may be highlighted by visual highlight indicia, such as a conspicuous color, e.g., outlined in yellow, indicating a
10 possible communication problem.

Fig. 45 is an example visual representation 760 of routes. A PAN 762 comprises devices 764, 766, 768, 770, 772 and 774 associated as shown. Routes 776, 778, 780 and 782 are represented by splines with terminating arrowheads. Route 776 defines a route from device 774 to device 764 to device 766 to device 772. Route 778 defines a route from device
15 766 to 764. Route 780 defines a route from device 764 to device 772, and route 782 defines a route from device 772 to 774. Each spline may be rendered with differentiating indicia, such as a particular color for each spline.

Fig. 46 is an example visual representation 790 of a selected device 774 and corresponding routes 776 and 782. Selecting the device 774 highlights the device and
20 displays only the routes to and from the device 774, i.e., routes 776 and 782. Multiple devices can be selected using a ctrl-select by clicking the left mouse button on each device while holding down the ctrl-key. Once selected, only those routes that begin or end on the selected device(s) are shown.

Fig. 47 is an example visual representation 800 of route selection. Upon selecting the
25 route 782, devices 772 and 774 are highlighted. In one embodiment, when a route is selected, each of the devices the route traverses will be highlighted by visual highlight indicia using the same color as the route spline color.

Fig. 48 is an example visual representation 810 of direct addressing in an APS layer, and Fig. 49 is an example visual representation 830 of indirect addressing in an APS layer.
30 The APS layer information for a PAN 812 having devices 814, 816, 818, and 820 may be displayed via the visual representation of the application endpoints 822, 824, 826 and 828.

The APS layer analysis is based on the detection of APS data packets. In a ZigBee network, the APS packet header includes the source endpoint, Profile ID, cluster ID, and destination endpoint.

There are two addressing schemes available in the APS layer - direct addressing and indirect addressing. In direct addressing, APS Packet Flows between end devices are identified. These packet flows may be broken down by end-point pairs and then be further broken down by profile ID and cluster ID. When a new packet flow is detected between endpoints, the visualization 810 is updated to show the endpoints on each end device, and to add a link between the source and destination endpoints to represent the binding. In one embodiment, the binding is implicitly determined by the network analysis system 10 by determining the link based on the flow of APS data packets and thus need not be based on the detection of explicit binding requests. The link is represented as a line linking the endpoints, such as the binding line 830 representing the binding between endpoints 822 and 828. Once selected, information about the binding, including the Profile ID and the list of cluster IDs, may be shown directly under the selected binding.

Selection of an endpoint may act as a filter that results in the display of only those bindings associated with this endpoint. If a single binding exists for an endpoint, selecting an endpoint will select the corresponding source and destination end devices for showing routing information. The selected endpoints, devices and binding are then highlighted. For example, selecting endpoint 822 in Fig. 48 will select and highlight devices 814 and 818, endpoints 822 and 828, binding line 830, and route 832. Profile IDs and the list of cluster IDs (incoming and outgoing) active on the selected endpoint may also be displayed.

If a binding does not exist, the selection of an endpoint may act as a filter whereby only future bindings that include this endpoint will be shown. Thus when a new binding is added, the corresponding source and destination devices are automatically selected for drawing routes.

When using indirect addressing, the source or destination endpoint may be absent from the packet. Indirect addressing is used when a particular end device does not have direct support for binding, i.e., the device does not know the final destination for the APS packets it sends. Instead, the end device will forward the APS packets to the coordinator, and the coordinator performs a binding table lookup to forward the packets to the associated destinations.

Indirect addressing is visually represented by considering the flow of packets from the source endpoint to the coordinator and the flow of packets from the coordinator to destination endpoints. Each segment is shown independently in the visual devices tree, and there is an INDIRECT endpoint object associated with the coordinator. All indirect bindings are shown intersecting with the indirect endpoint object. Packet flows from source endpoint to the coordinator are shown as a binding from the source endpoint to the INDIRECT endpoint, and

packet flows from the coordinator to destination endpoints are shown as a binding from the INDIRECT endpoint to the destination endpoints.

Selecting a source endpoint results in automatic selection of the corresponding end device and route to the coordinator. For example, in Fig. 49, selecting endpoint 826 results in the selection and highlighting of the endpoint 826 and the indirect endpoint 824, the binding line 842, and routes 832, 834, 836, 838, and 844.

The visual device tree also provides context menus to quickly access functions relative to various displayed objects. The right-click menu item can be selected after moving a mouse above an object in the visual device tree and selecting the right-click menu button. Device context menus, route context menus, APS endpoint context menus, and APS binding context menus similar to the context menus previously described may thus be readily accessed by a user.

Fig. 50 is a visual device layout window 900. The visual device layout window 900 facilitates an alternate view of a visual device tree. Instead of the network analysis system automatically drawing devices in a tree, the visual device layout window 900 provides for devices to be placed by the user on a user-defined background image 906, such as a floor plan of an office. This allows the user to create a visual representation that corresponds to the physical layout of the network as deployed in the office.

Network objects are initially displayed in a placement pane 902. The placement pane 902 initially contains all identified devices and any new devices that join the network. In one embodiment, devices may be placed in a device naming table. Devices in the device naming table are displayed in the placement pane 902 and can be placed on the background image regardless of whether the devices are part of the current identified network. This allows devices to be placed on a background image to represent their physical location prior to the network being formed.

Devices may be placed in the background image by conventional drag-and-drop operations. As illustrated in Fig. 50, device objects 914, 916 and 918 have been placed against a background image 906, representing the placement of the corresponding devices as deployed in an office. Device object 912 is to be placed in area 920 on the background image 906, as represented by the dashed arrow.

Fig. 51 is the visual device layout window 900 showing associations, routes, endpoints and bindings. The physical deployment of the network corresponding to the PAN object 910 is thus represented by the device objects 912, 914, 916 and 918. Endpoint bindings 922, 924, 926, 928, 930, 932 and 934 are also displayed, as are associations, routes

and binding lines. The association, route and binding lines may be selectively displayed to reduce clutter.

Although Figs. 50 and 51 depict a two-dimensional representation, three-dimensional representations may also be used. A three-dimensional representation may be used to visually represent a network deployed on two or more floors, or in a large space such as a warehouse. Fig. 52 depicts an example three-dimensional representation 940.

In another embodiment, endpoints are visually represented as intrinsic elements of a network device object, and endpoint bindings are drawn between nodes. Fig. 53 is an example visual representation 950 of a network topology having endpoints that are visually represented as intrinsic elements of network device objects. A straight arrow is drawn between device objects to represent detected APS binding between the device objects. The arrow may be unidirectional when only an APS binding in one direction is detected, and may be bidirectional when an APS binding is detected in both directions. The intrinsic representation may also be used in rendering the visual representation of the network topology without reference to a visual device layout.

The steps and the order of the steps in the methods and flowcharts described herein may be altered, modified and/or augmented and still achieve the desired outcome. Additionally, the methods, flow diagrams and structure block diagrams described herein may be implemented in the example processing devices described herein by program code comprising program instructions that are executable by the device processing subsystem. Other implementations may also be used, however, such as firmware or even appropriately designed hardware configured to carry out the methods and flow diagrams or implement the structure block diagrams described herein. Additionally, the methods, flow diagrams and structure block diagrams that describe particular methods and/or corresponding acts in support of steps and corresponding functions in support of disclosed software structures may also be implemented in software stored in a computer readable medium and equivalents thereof. The software structures may comprise source code, object code, machine code, or any other persistently or temporarily stored code that is operable to cause one or more processing systems to perform the methods described herein or realize the structures described herein.

This written description sets forth the best mode of the invention and provides examples to describe the invention and to enable a person of ordinary skill in the art to make and use the invention. This written description does not limit the invention to the precise terms set forth. Thus, while the invention has been described in detail with reference to the examples set forth above, those of ordinary skill in the art may effect alterations,

modifications and variations to the examples without departing from the scope of the invention.

~~What is claimed is:~~
What is claimed is:

1. A packet-based network analysis system, comprising:

a correlator processor configured to:

receive packet records corresponding to packets communicated over a network

5 and store the packet records in a data store;

determine if a received packet record corresponds to a retransmitted packet or a duplicate packet by comparing the received packet record to the packet records stored in the data store;

10 if the received packet record corresponds to a retransmitted packet based on the comparison of the received packet record to a packet record stored in the data store, then discard the received packet record and classify the packet record stored in the data store as a retransmitted packet record;

15 if the received packet record corresponds to a duplicate packet based on a comparison of the received packet record to a packet record stored in the data store, then timestamp synchronize the packet record stored in the data store and the received packet record and discard the received packet record; and

generate correlated packet records from the packet records stored in the data store, the correlated packet records representative of the order in which the packets were transmitted in the network.

20

2. The packet-based network analysis system of claim 1, further comprising:

a capture device configured to monitor packets communicated over the network and create the corresponding packet records.

25 3. The packet-based network analysis system of claim 2, wherein:

the capture device is configured to create corresponding packet records by including a subset of a packet in each corresponding packet record.

4. The packet-based network analysis system of claim 1, wherein:

30 the correlator processor is configured to compare the received packet record to the packet records stored in the data store by comparing packet data in the received packet record to packet data of the packet records stored in the data store.

5. The packet-based network analysis system of claim 4, wherein:

the correlator processor is configured to compare packet data in the received packet record to packet data of the packet records stored in the data store by generating cyclic redundancy checksums for the packets.

5 6. The packet-based network analysis system of claim 1, further comprising:
a plurality of capture devices configured to monitor packets communicated over the network and create the corresponding packet records, and wherein each capture device includes a capture clock and each capture device is further configured to include a timestamp in each packet record corresponding to the capture clock time the capture device detects the
10 start of frame of a packet.

7. The packet-based network analysis system of claim 6, wherein:
the correlator is configured to timestamp synchronize the packet records stored in the data store and the received packet records by designating a first capture device as a master
15 capture device and time synchronize packet records received from other capture devices to the master capture device.

8. A packet-based network analysis system for analyzing packet flow among devices in a wireless network, comprising:
20 a wireless network analysis processing device configured to:
receive a plurality of correlated packet records, the correlated packet records representative of the order in which corresponding packets were transmitted in the network and including media access control (MAC) layer data and network layer data for each corresponding packet;
25 process the MAC layer data and network layer data to generate network topology data representative of the network topology, generate packet flow data representative of the flow of packets between devices at the MAC layer and across the network at the network layer, and generate measurement data relating to the packet flow data.

30 9. The packet-based network analysis system of claim 8, wherein:
the wireless network analysis processing device is configured to:
identify packet streams based on a source network layer address and a destination network layer address.

35 10. The packet-based network analysis system of claim 9, wherein:

the wireless network analysis processing device is configured to:

compare a source MAC address and a destination MAC address to the source network layer address and the destination network layer address to generate packet flow data for the packet streams.

5

11. The packet-based network analysis system of claim 8, wherein:

the packet flow data comprises packet flow records, and the wireless network analysis processing device is configured to:

10 create a new packet flow record and add packet flow data corresponding to a first hop of a packet in the network if the source MAC address matches the source network address in a corresponding correlated packet record;

add packet flow data corresponding to a last hop of a packet in the network to a packet flow record if the destination MAC address matches the destination network address in a corresponding correlated packet record; and

15 add packet flow data corresponding to an intermediate hop of a packet in the network to a packet flow record if the destination MAC address does not match the destination network address and the source MAC address does not match the source network address in a corresponding correlated packet record.

20

12. The packet-based network analysis system of claim 8, wherein:

the wireless network analysis processing device is configured to:

process application layer data to classify endpoints in the network topology.

13. The packet-based network analysis system of claim 12, wherein:

25

the network comprises a ZigBee network; and

the endpoints comprise source endpoints and destination endpoints, and the wireless network analysis processing device is configured to generate endpoint binding data for related source endpoints and destination endpoints.

30

14. The packet-based network analysis system of claim 8, further comprising:

a correlator processor configured to:

receive packet records corresponding to packets communicated over the network and store the packet records in a data store; and

35 generate the correlated packet records from the packet records stored in the data store, the correlated packet records comprising hop data and timing data representative

of the packet path and packet transmission times in the network for each corresponding packet, the correlated packet records filtered of packet records for duplicate packet transmissions and packet retransmissions.

- 5 15. A packet-based wireless network visualization system, comprising:
a data store;
an input/output subsystem including a display device; and
a processing subsystem in data communication with the data store and the
input/output subsystem, the packet-based wireless network visualization system configured
10 to:
receive network topology data, packet flow record data, and measurements data over
the input/output subsystem and store the network topology data, packet flow record data, and
measurements data in the data store;
generate a visual representation of a network topology on the display based on the
15 network topology data, the visual representation of the network topology including device
objects and associations of device objects, the visual representation comprising a plurality of
layer representations;
generate a visual representation of packet flows within the network topology based on
the packet flow records; and
20 selectively display measurement data related to the packet flows and network
topology based on the measurements data.

- 25 16. The packet-based wireless network visualization system of claim 15, wherein:
the plurality of layer representations include a media access control (MAC) layer and
a network layer.

- 30 17. The packet-based wireless network visualization system of claim 16, wherein:
the plurality of layer representations includes an application support (APS)
layer.

- 35 18. The packet-based wireless network visualization system of claim 17, wherein:
the wireless network comprises a ZigBee network, and the packet-based wireless
network visualization system is further configured to:
generate a visual representation indicating a binding between application endpoints.

19. The packet-based wireless network visualization system of claim 18, wherein the packet-based wireless network visualization system is further configured to:

generate a visual representation of device functions associated with the application endpoints.

5

20. A computer-implemented method of visualizing a packet-based wireless network, comprising:

generating a visual representation of a network topology, the visual representation of the network topology comprising device objects and associations of device objects and a plurality of layer representations;

10

generating a visual representation of packet flows within the network topology;

applying visual highlight indicia to the visual representation of packet flows associated with a device object upon the selection of a device object; and

applying visual highlight indicia to the device objects upon the selection of a visual representation of a packet flow associated with the device object.

15

21. The method of claim 20, wherein:

the wireless network comprises a ZigBee network, and further comprising:

generating a visual representation indicating a binding between application endpoints;

20

generating a visual representation of device functions associated with the application endpoints.

22. The method of claim 20, wherein:

applying visual highlight indicia to the visual representation of packet flows associated with a device object upon the selection of a device object comprises outlining

25

comprises:
highlighting visual representations of packet flows for which the device objection is representative of a source device or destination device of the visual representations of packet flows; and

30

applying visual highlight indicia to the device objects upon the selection of a visual representation of a packet flow associated with the device object comprises:

highlighting all device objects intersected by the selected visual representation of packet flows.

35

23. A packet-based wireless network visualization system, comprising:

a correlator subsystem configured to:

receive packet records corresponding to packets communicated over a network and store the packet records in a data store;

5 determine if a received packet record corresponds to a retransmitted packet or a duplicate packet by comparing the received packet record to the packet records stored in the data store;

10 if the received packet record corresponds to a retransmitted packet based on the comparison of the received packet record to a packet record stored in the data store, then discard the received packet record and classify the packet record stored in the data store as a retransmitted packet record;

if the received packet record corresponds to a duplicate packet based on a comparison of the received packet record to a packet record stored in the data store, then timestamp synchronize the packet record stored in the data store and the received packet record and discard the received packet record; and

15 generate correlated packet records from the packet records stored in the data store, the correlated packet records representative of the order in which the packets were transmitted in the network;

a wireless network analysis subsystem configured to:

20 receive the plurality of correlated packet records, the correlated packet records representative of the order in which corresponding packets were transmitted in the network and including media access control (MAC) layer data and network layer data for each corresponding packet; and

25 process the MAC layer data and network layer data to generate network topology data representative of the network topology, generate packet flow data representative of the flow of packets between devices at the MAC layer and across the network at the network layer, and generate measurement data relating to the packet flow data; and

a wireless network visualization subsystem configured to:

30 receive network topology data, packet flow record data, and measurement data;

generate a visual representation of a network topology on the display based on the network topology data, the visual representation of the network topology including device objects and associations of device objects, the visual representation comprising a plurality of layer representations;

generate a visual representation of packet flows within the network topology based on the packet flow records; and

selectively display measurement data related to the packet flows and network topology based on the measurements data.

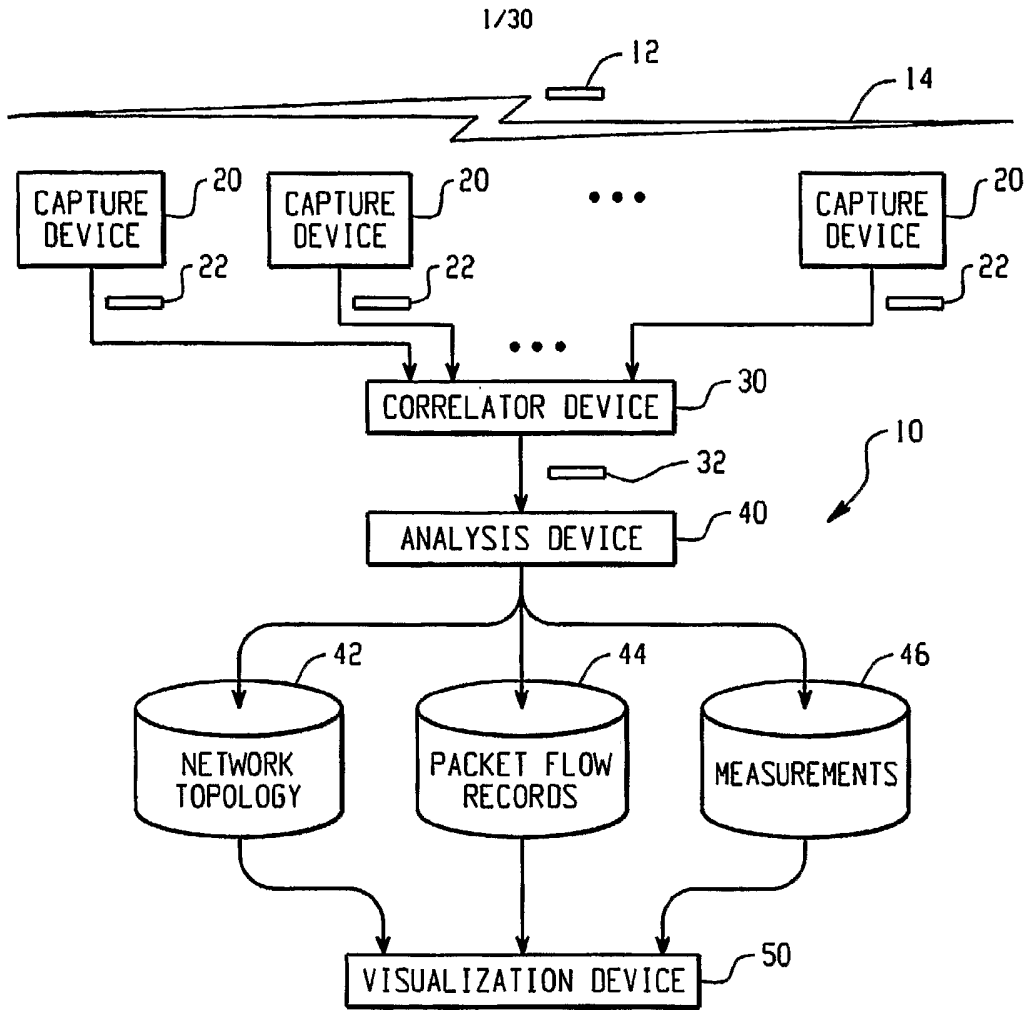


Fig. 1

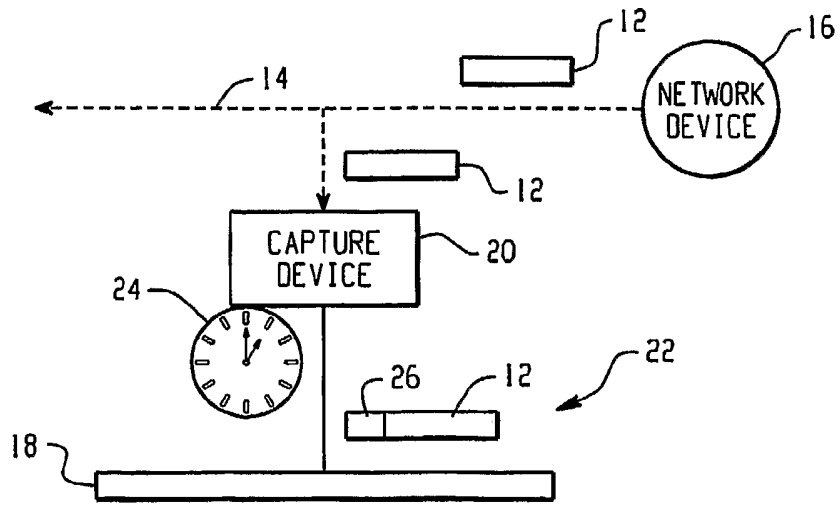


Fig. 2

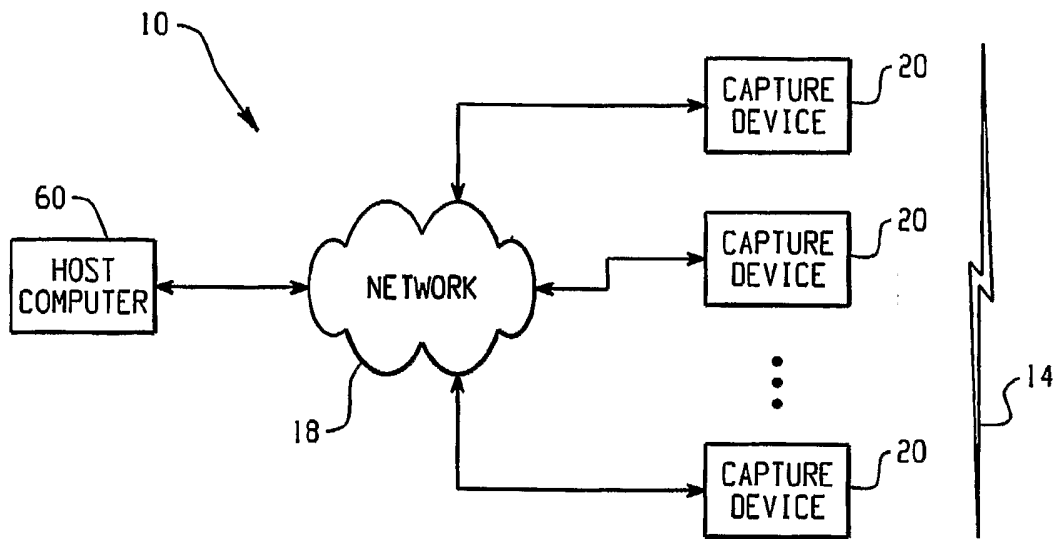


Fig. 3

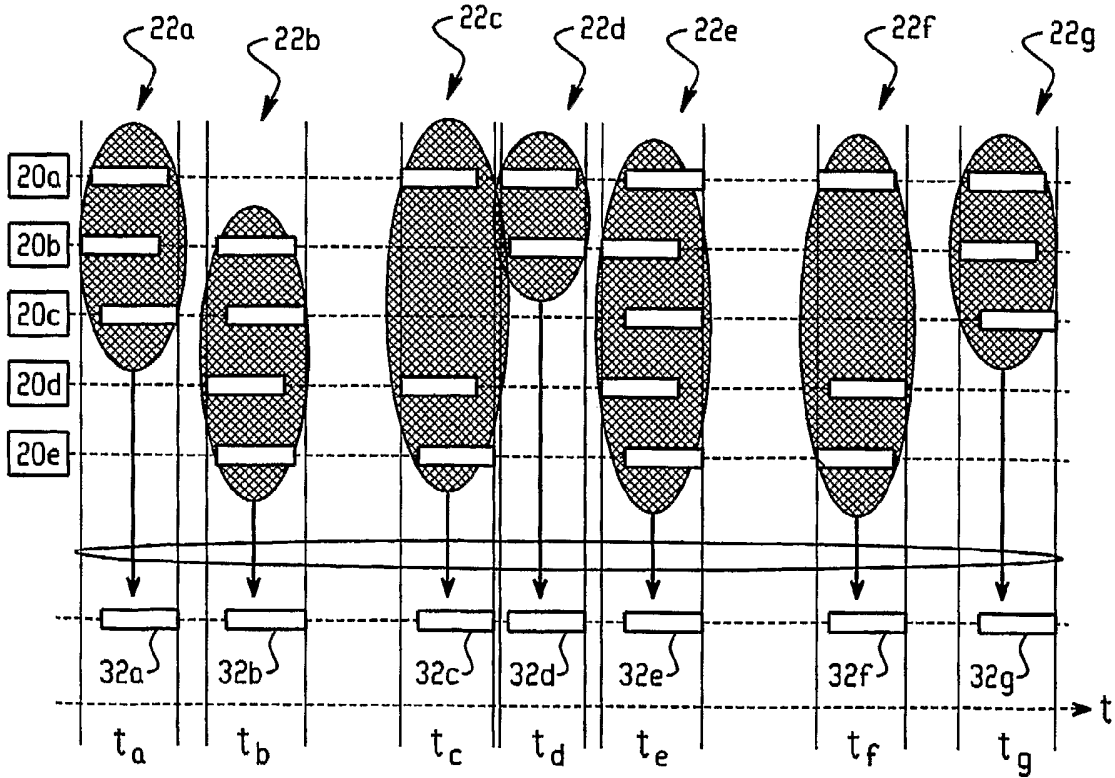


Fig. 4

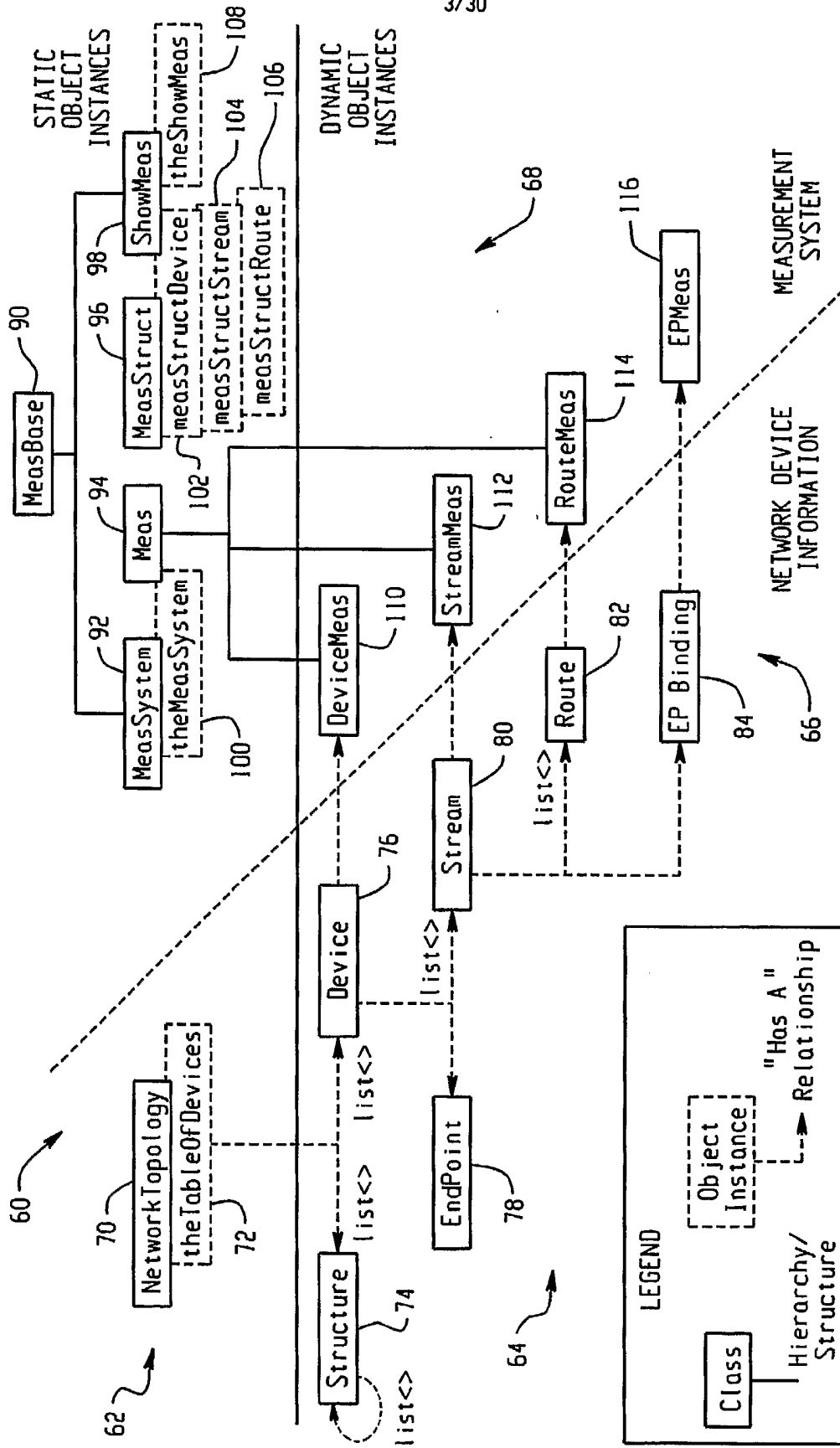


Fig. 5

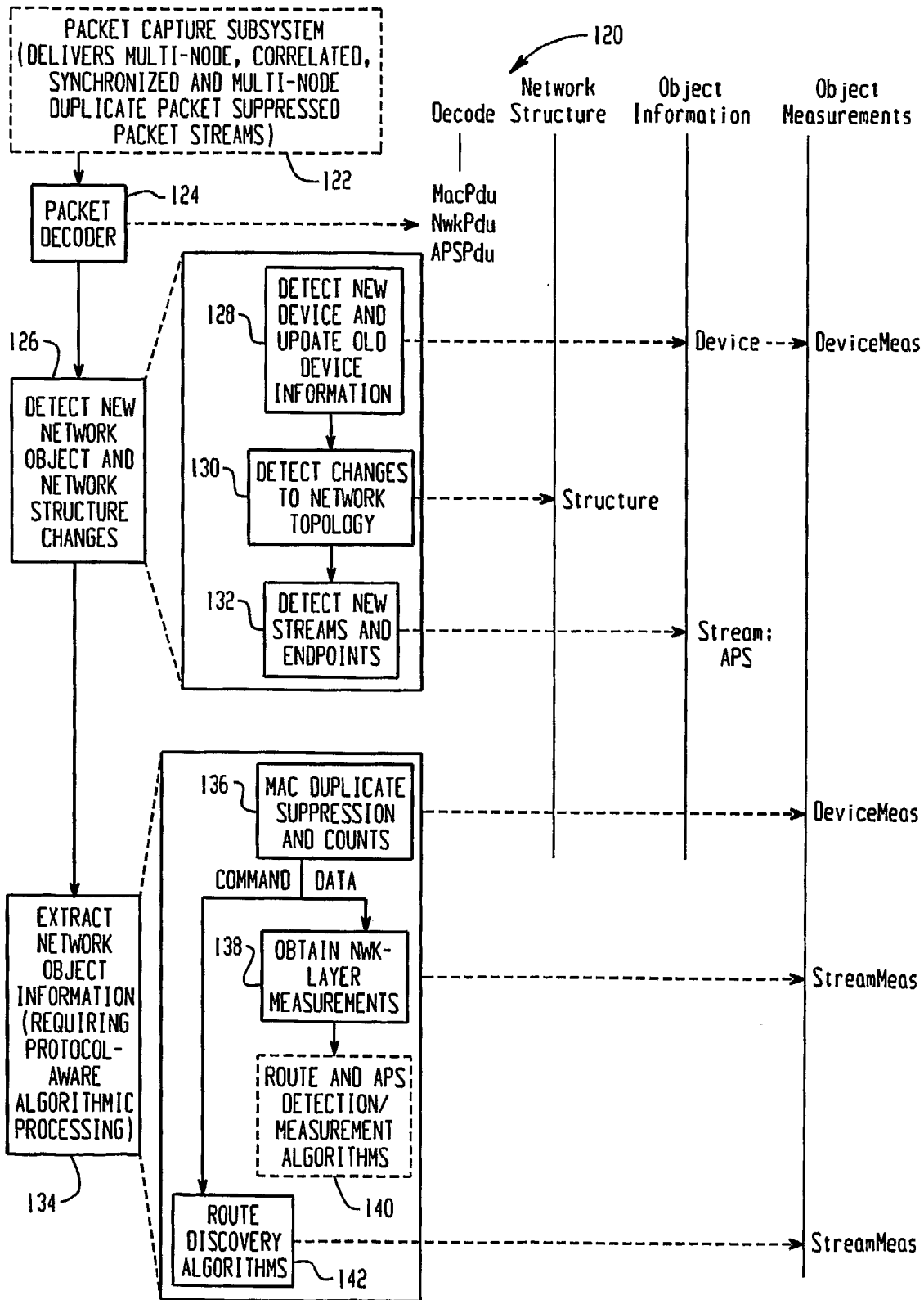


Fig. 6

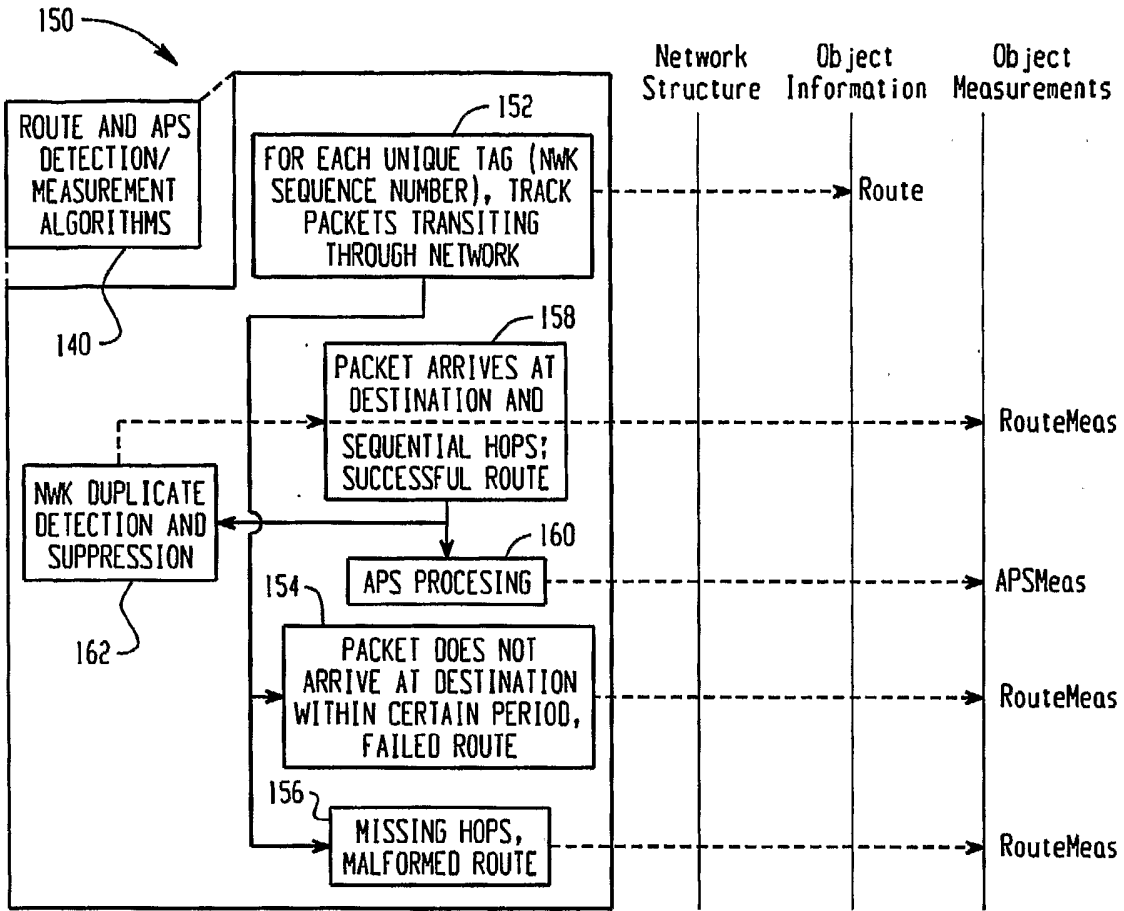


Fig. 7

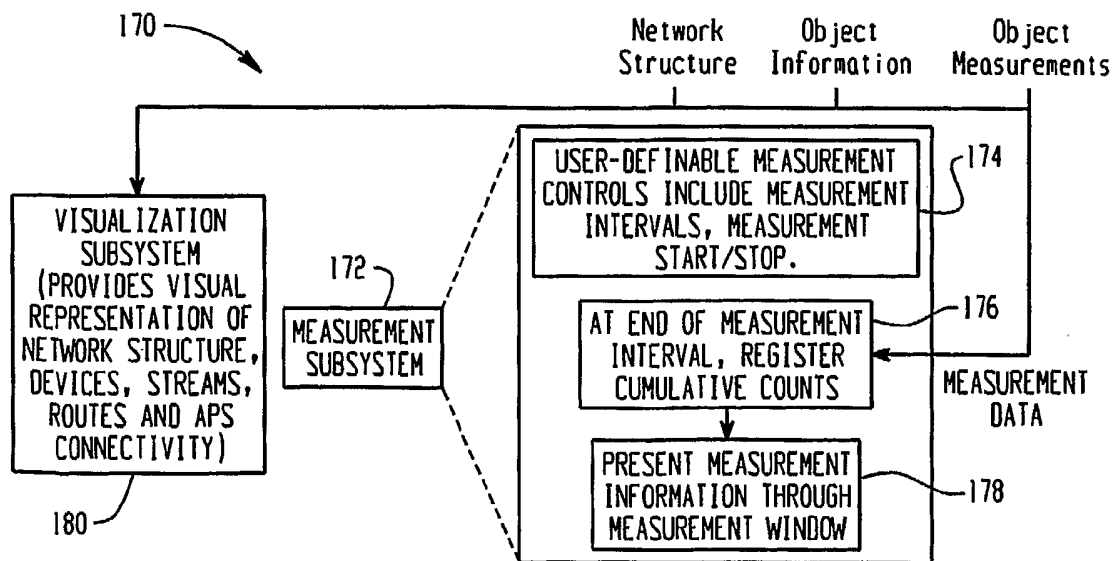


Fig. 8

6/30

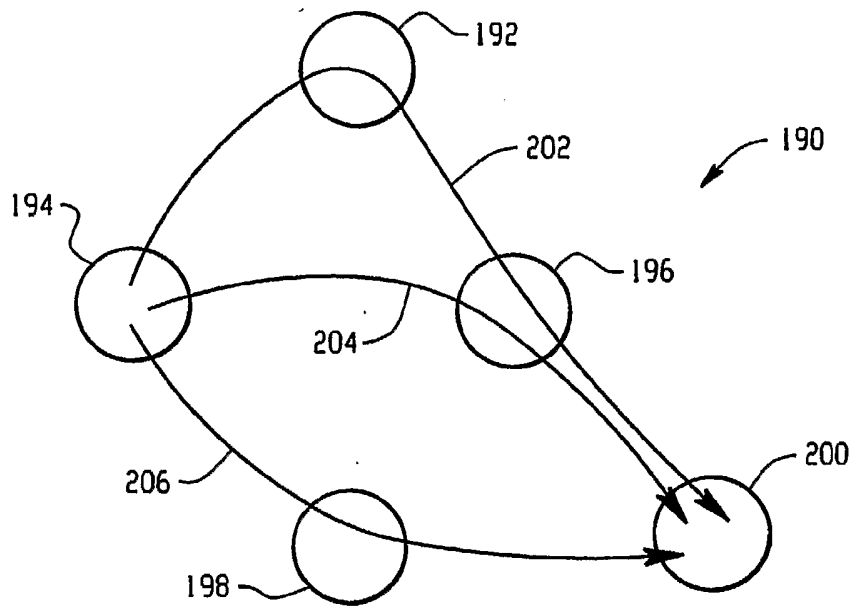


Fig. 9

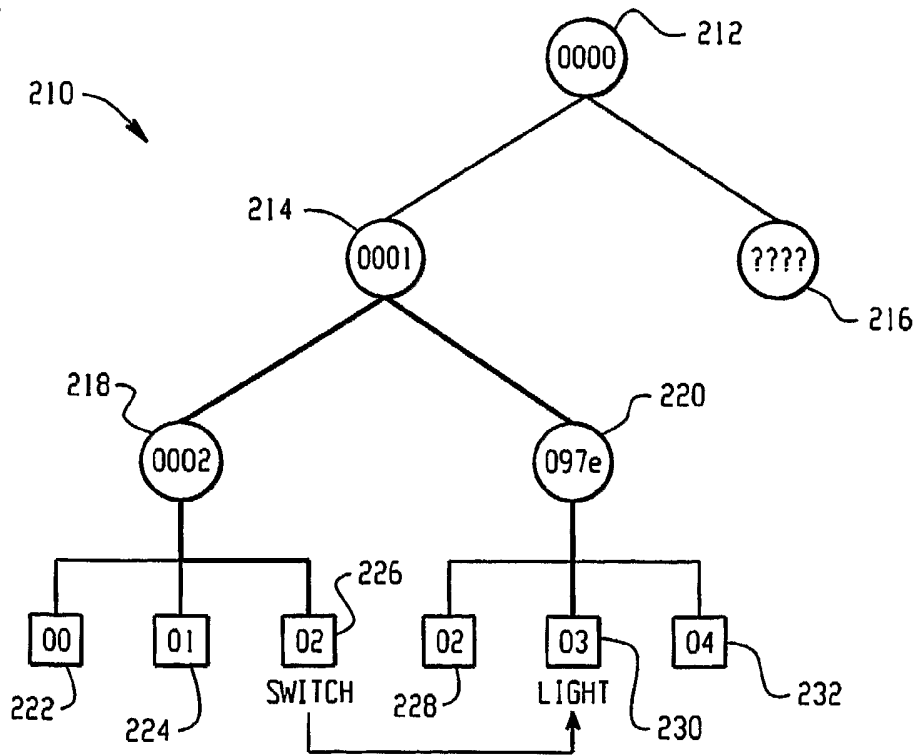


Fig. 10

7/30

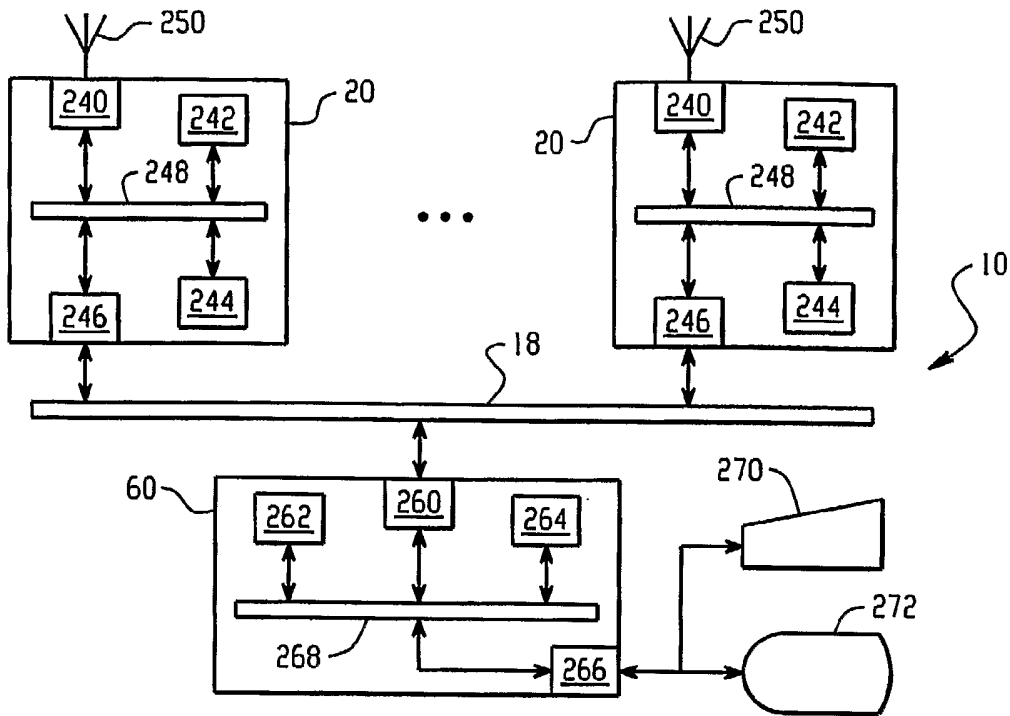
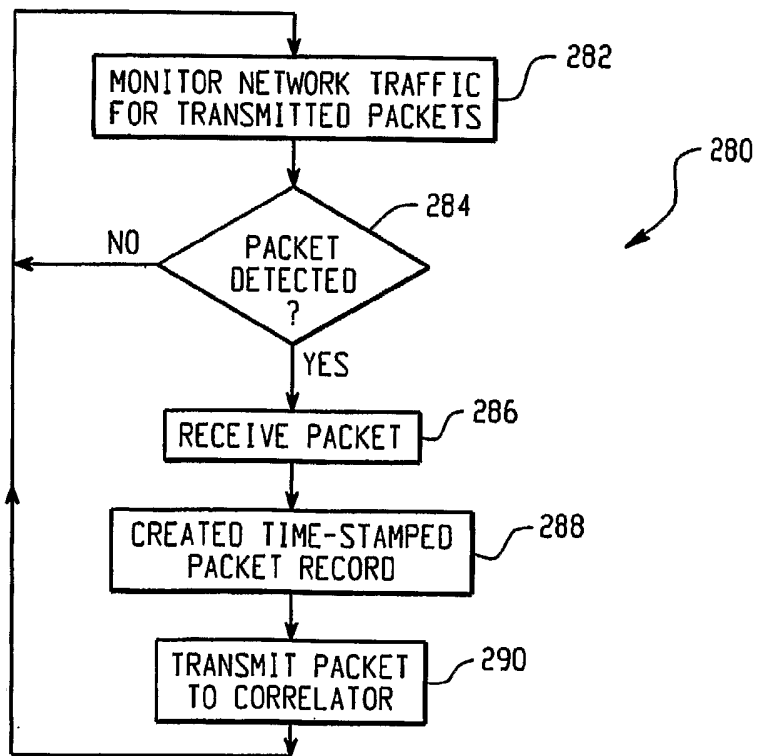


Fig. 11

Fig. 12



8/30

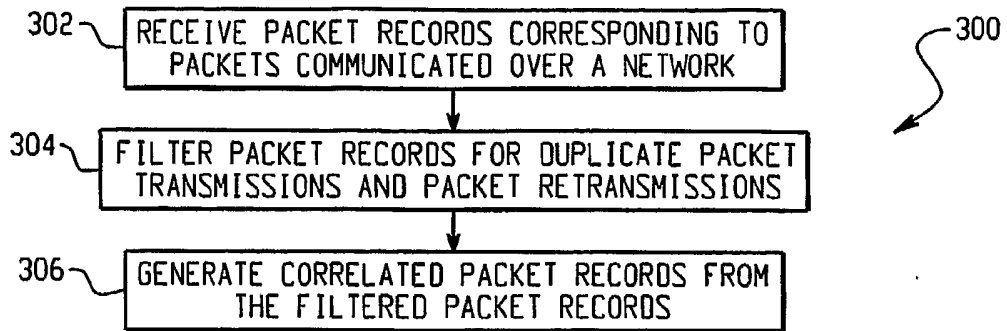


Fig. 13

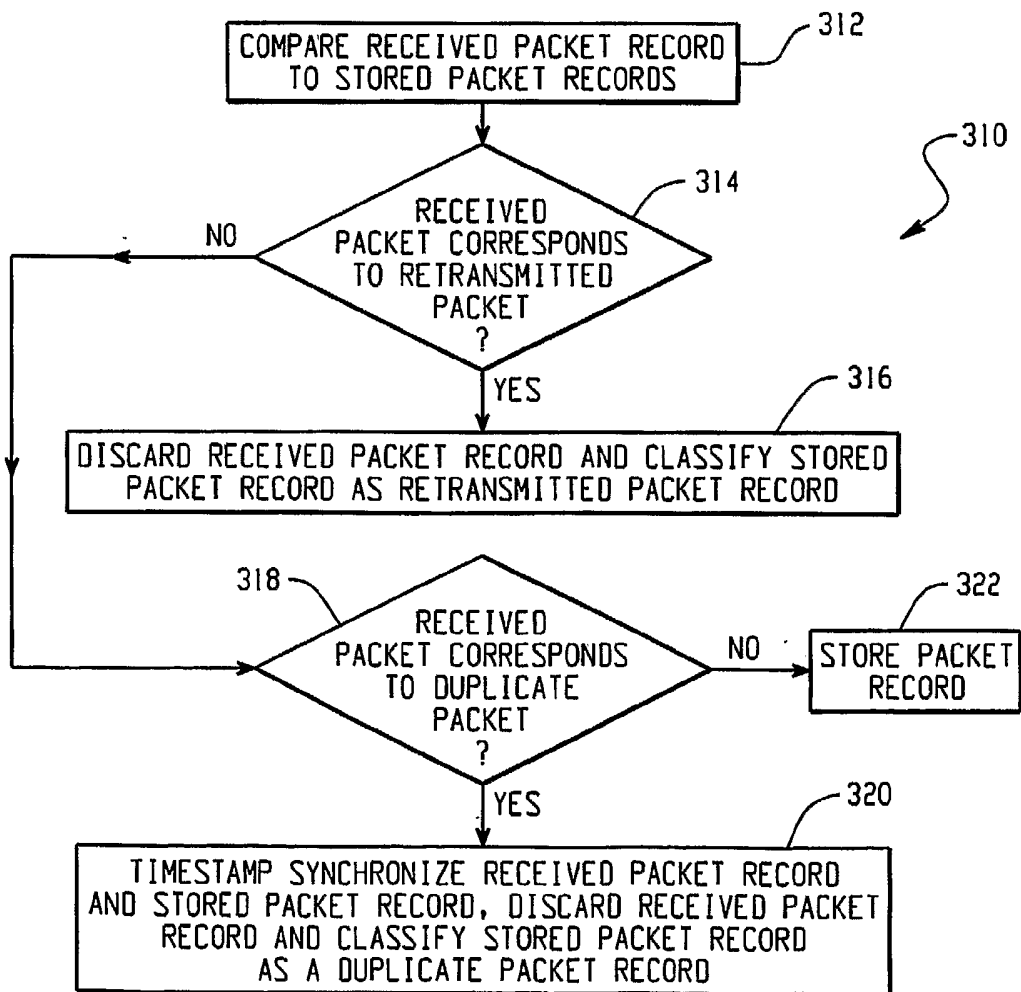


Fig. 14A

9/30

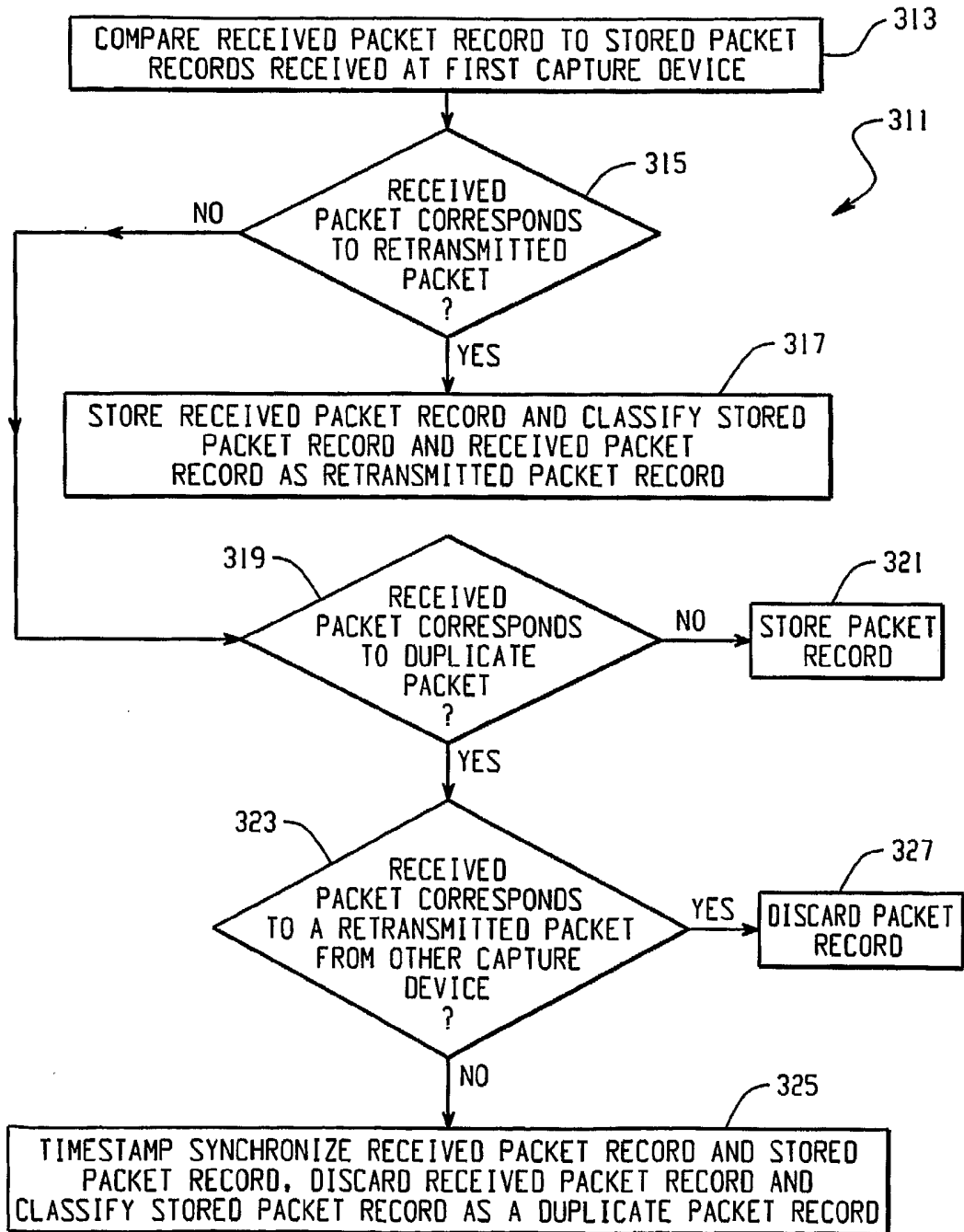


Fig. 14B

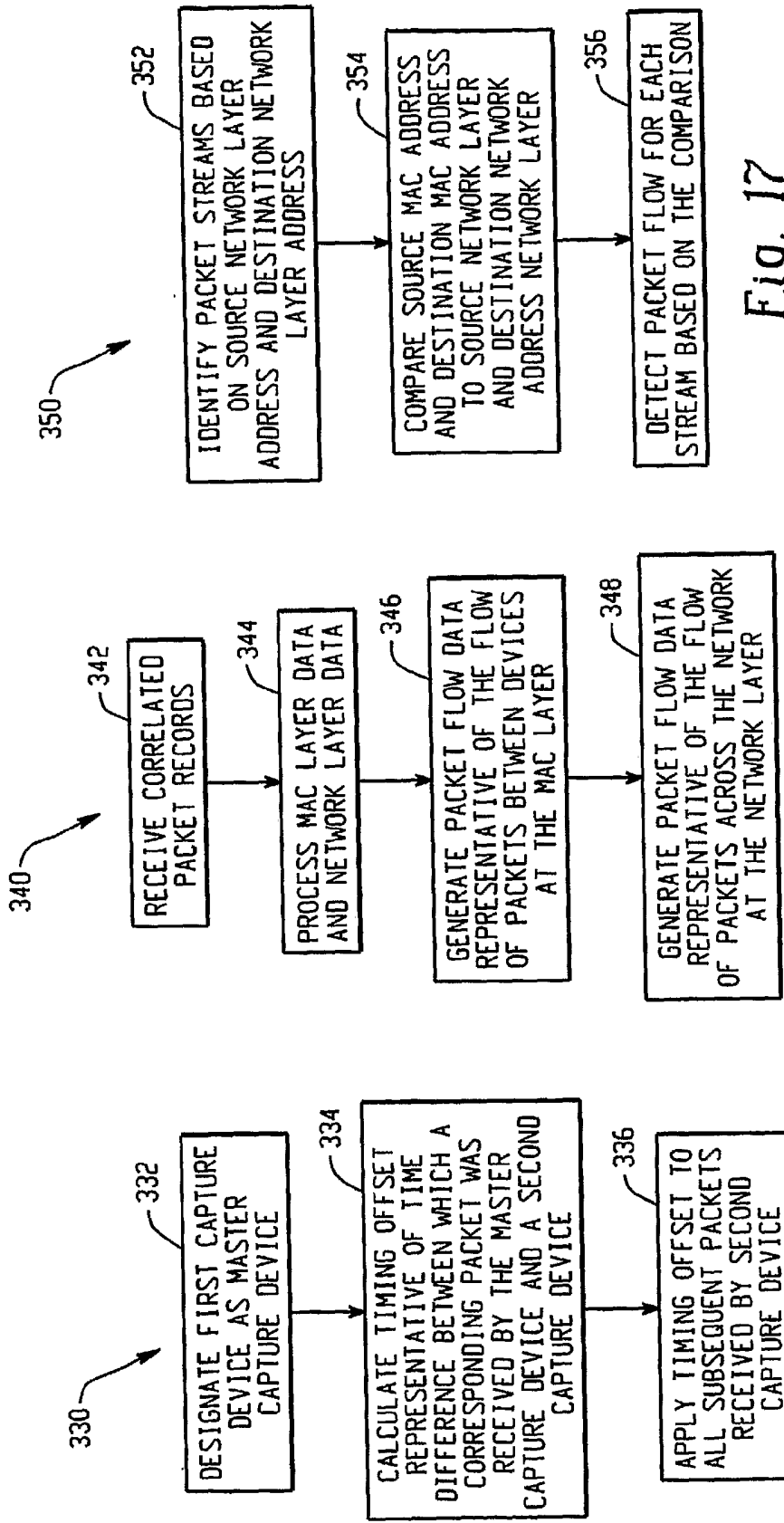


Fig. 17

Fig. 16

Fig. 15

11/30

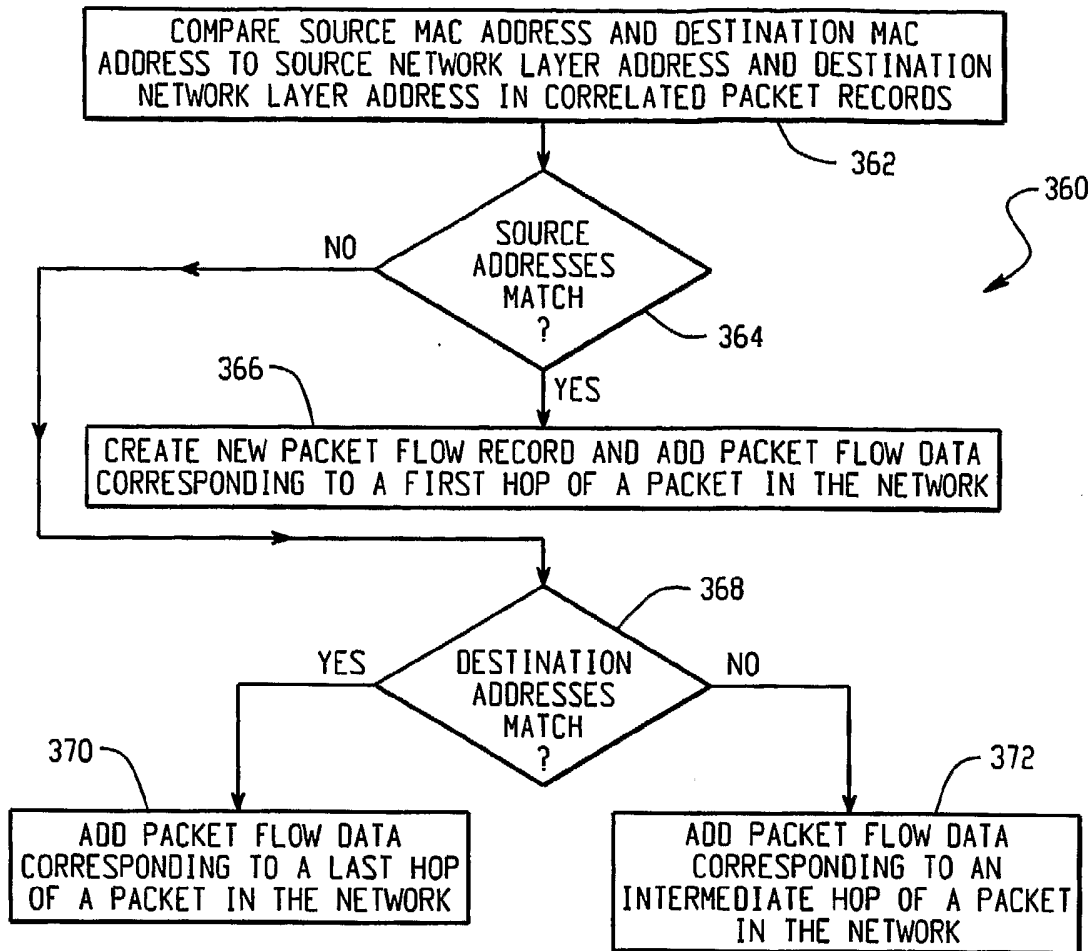


Fig. 18

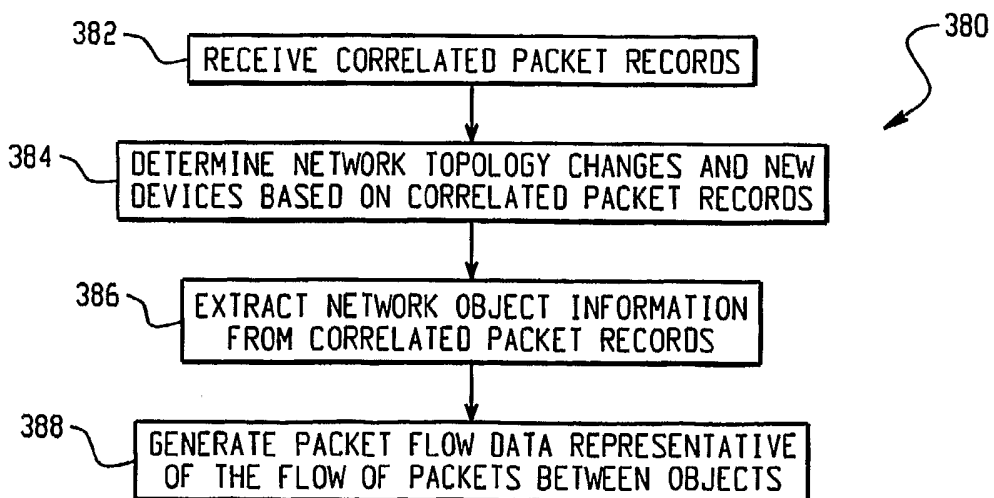


Fig. 19

12/30

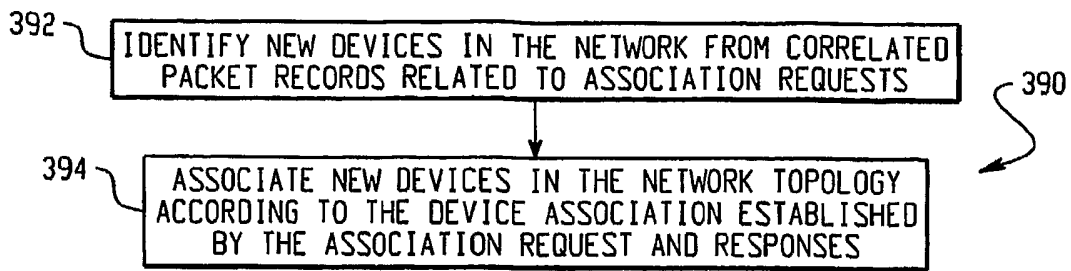


Fig. 20

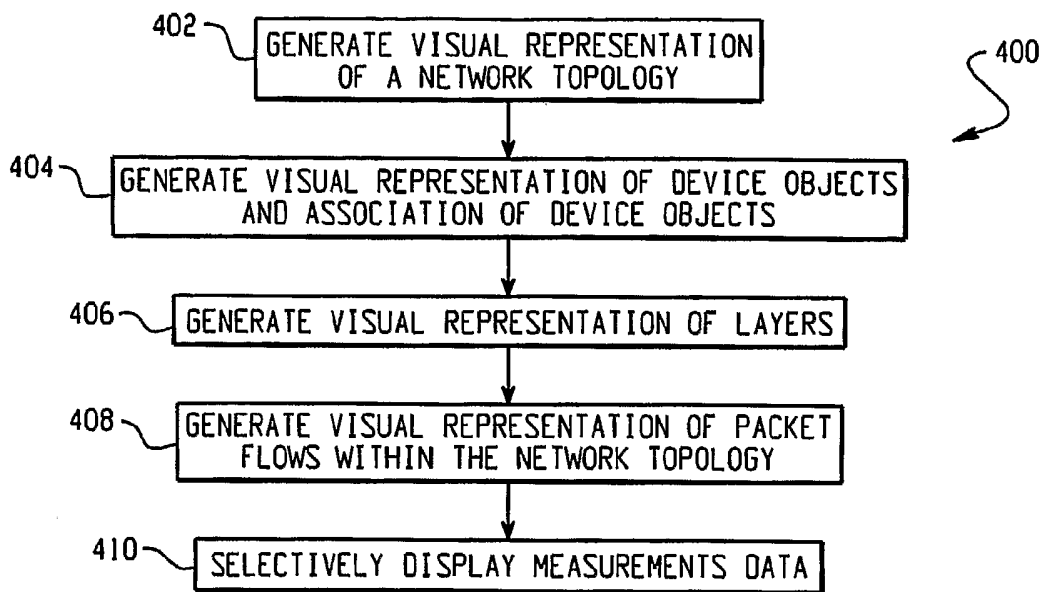


Fig. 21A

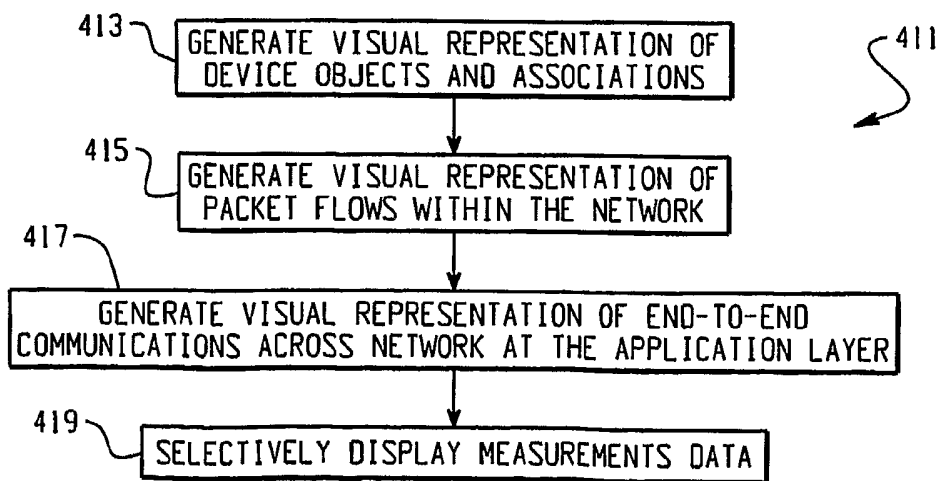


Fig. 21B

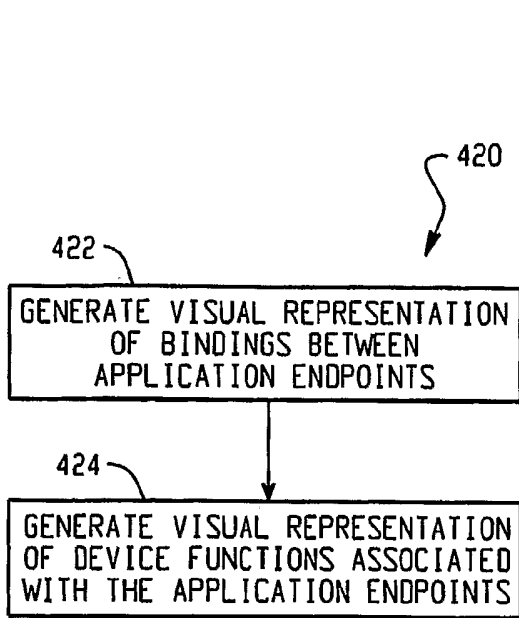


Fig. 22

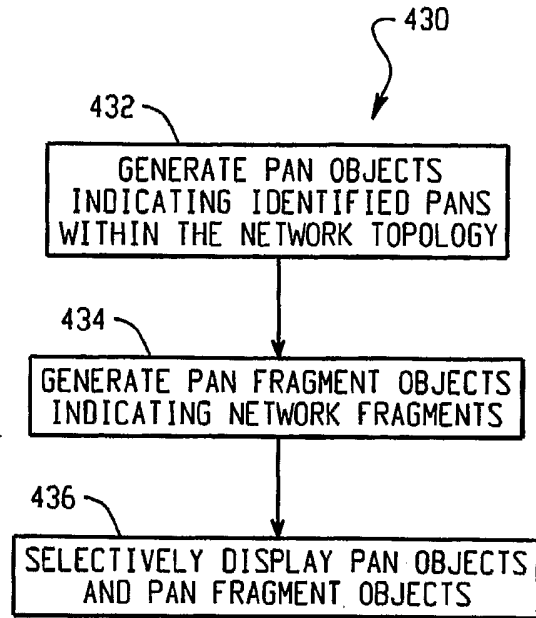


Fig. 23

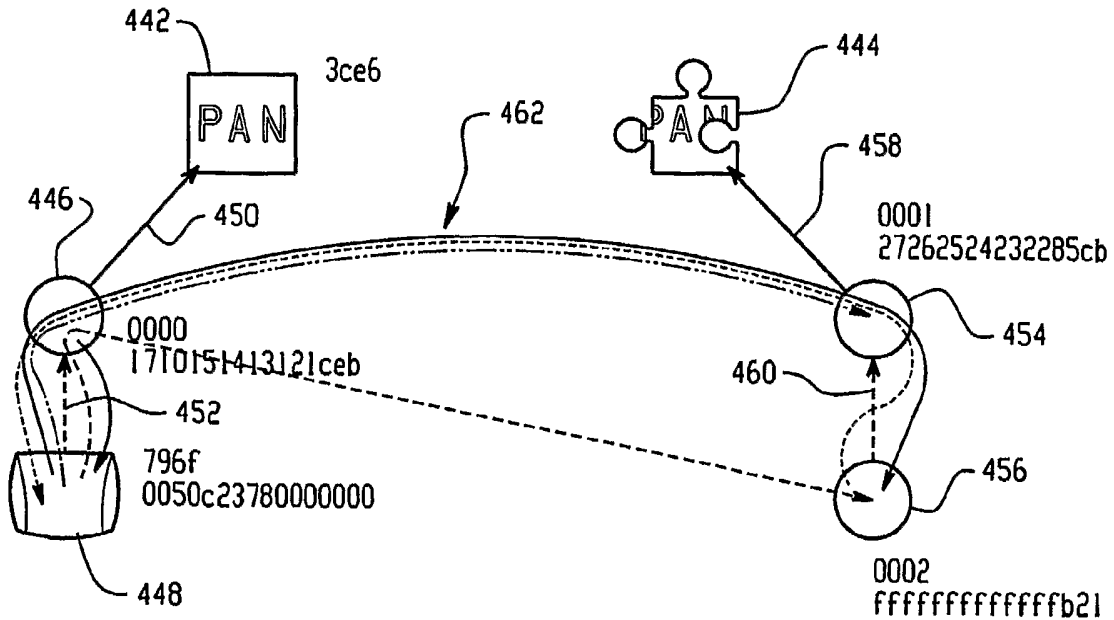


Fig. 24

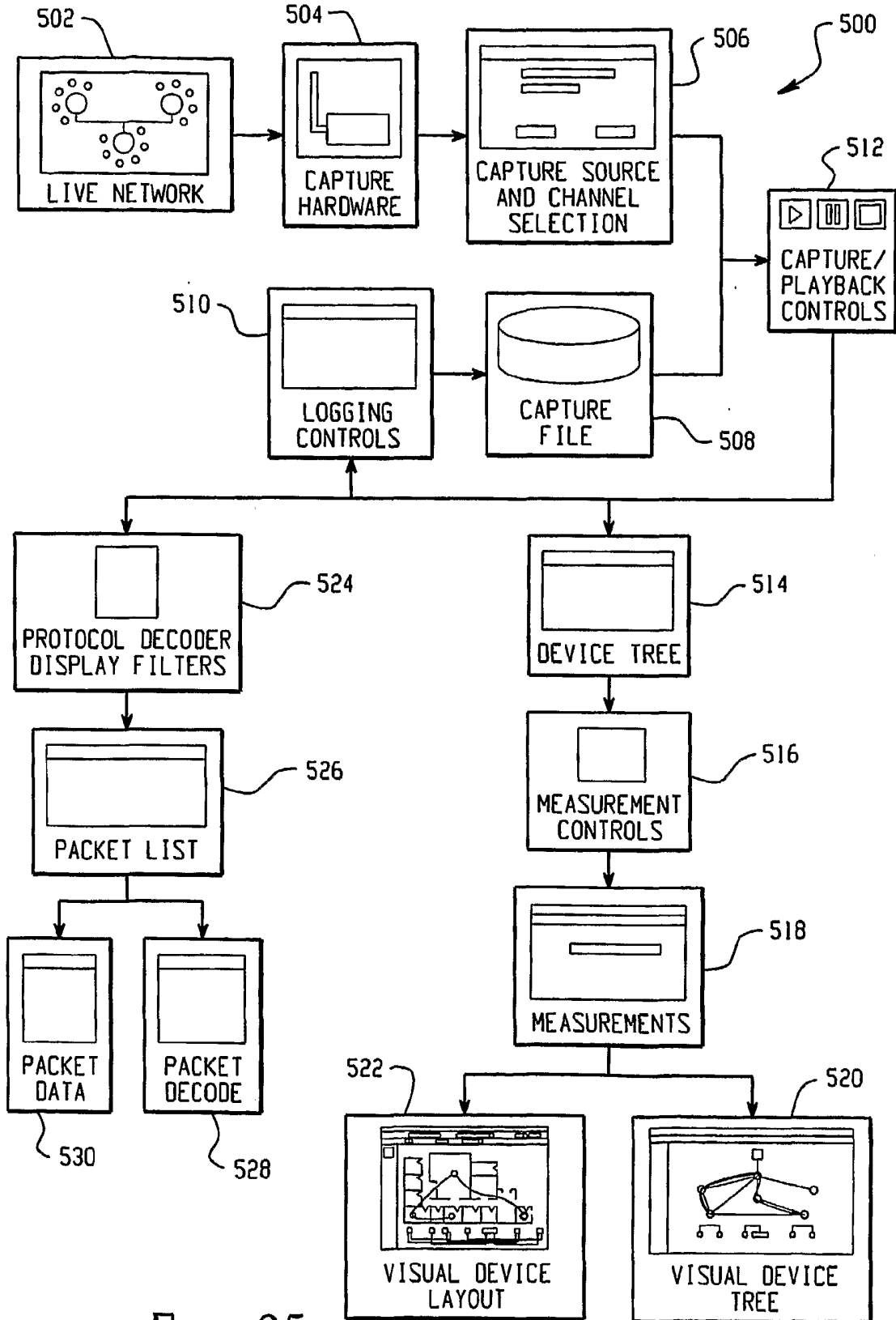


Fig. 25

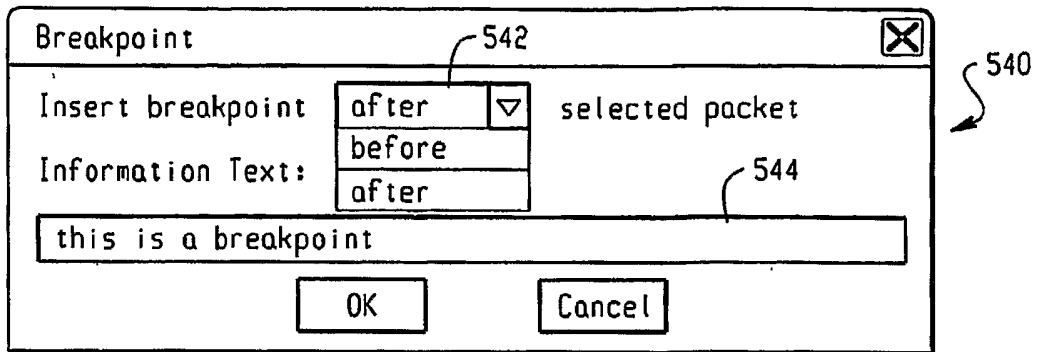


Fig. 26

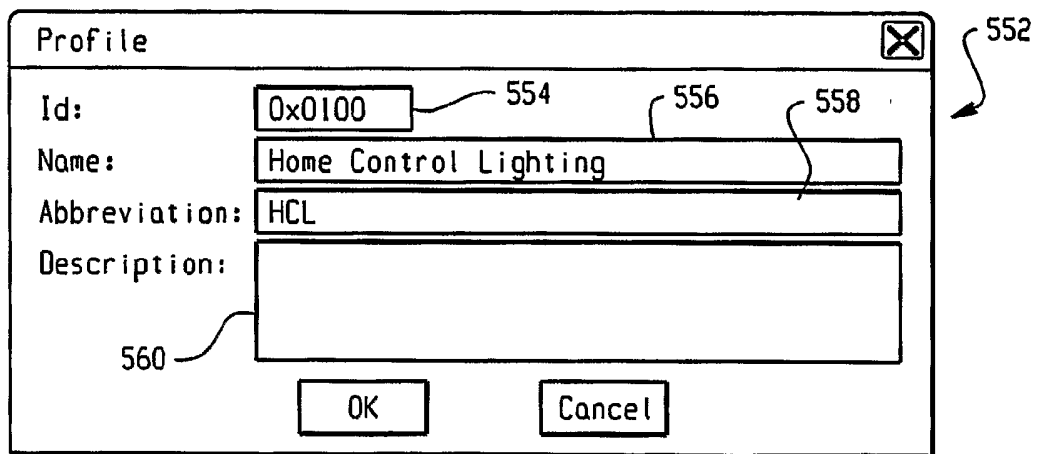


Fig. 28

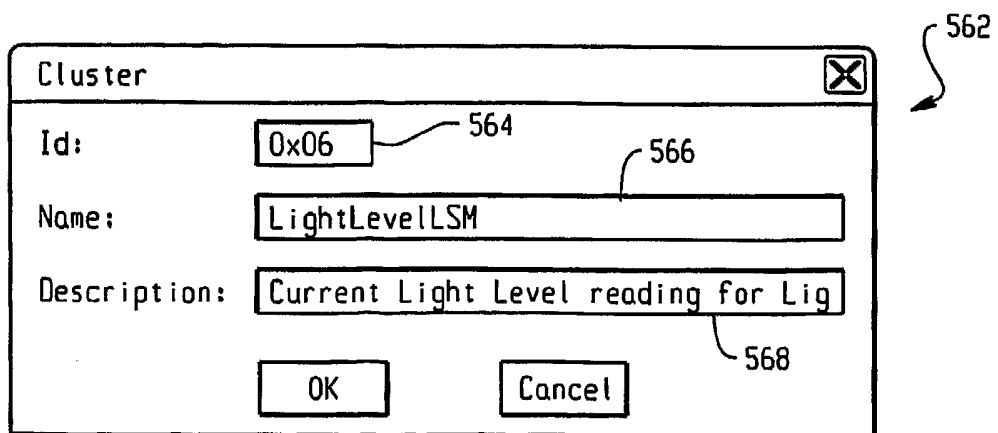


Fig. 29

✕
□ Daintree Profile Editor-Profile_HCL.xml

File Profile Clusters Help

Profile 0x0100

Name: Home Control Lighting

Abbreviation: HCL

Description:

Clusters

+
✎
✕

Id	Name	Description
0x13	OnOffSRC	ON or OFF Commands for Switch Remote Control
0x01	OnOffDRC	ON or OFF Commands for Dimmer Remote Control
0x02	DimBrightDRC	DIM or BRIGHT Commands for Dimmer Remote Control
0x03	PresetDRC	Presets Lighting level for Dimmer Remote Control
0x06	LightLevelLSM	Current Light Level reading for Light Sensor Monochromatic
0x08	OccupancyOS	Current Occupancy state for Occupancy Sensor
0x07	ProgramLSM	Program input to set up key parameters for the Light Sensor Monochromatic
0x14	ProgramSRC	Program input to set up key parameters for the Switch Remote Control
0x05	AdfDRC	Output setup and additional dimmer commands to external device
0x04	ProgramDRC	Program input to set up key parameters for the Dimmer Remote Control
0x09	ProgramOS	Program input to set up key parameters for the Occupancy Sensor
0x16	ProgramSLC	Program input to set up key parameters for the Switching Load Controller
0x15	StatusSLC	Load status and energy consumption for Switching Load Controller
0x12	ProgramDLC	Program input to set up key parameters for the Dimming Load Controller
0x11	StatusDLC	Load status and energy consumption for Dimming Load Controller

550 ↗

Fig. 27

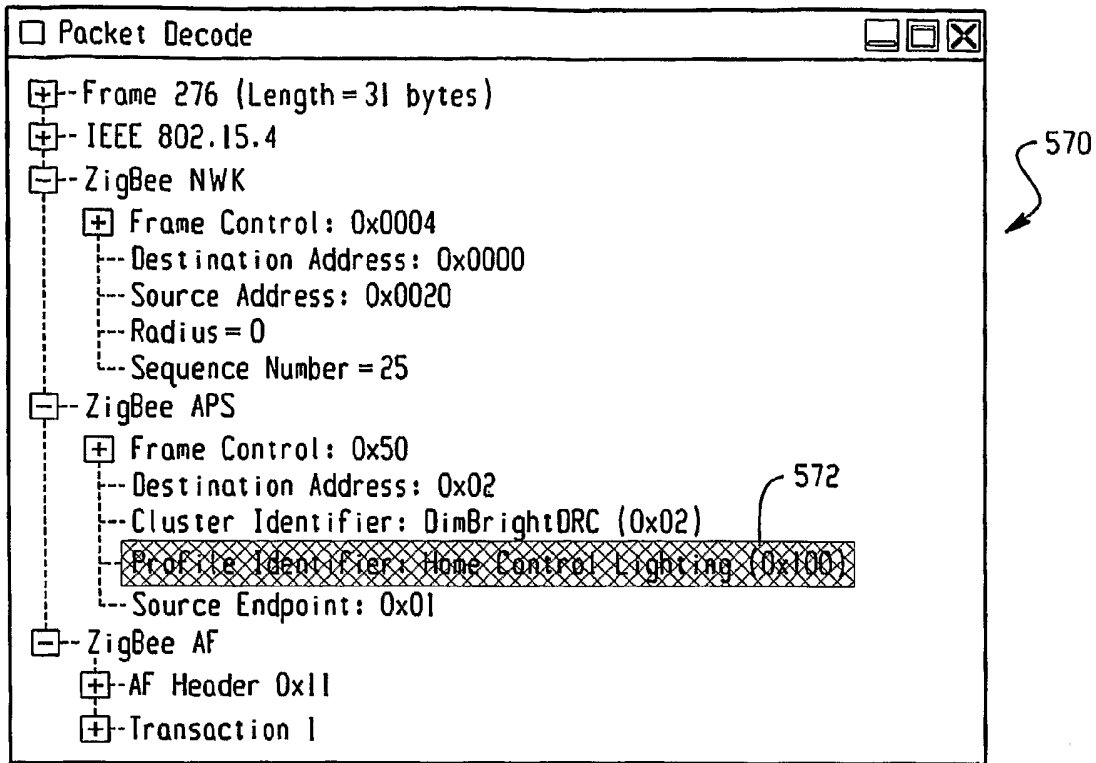


Fig. 30

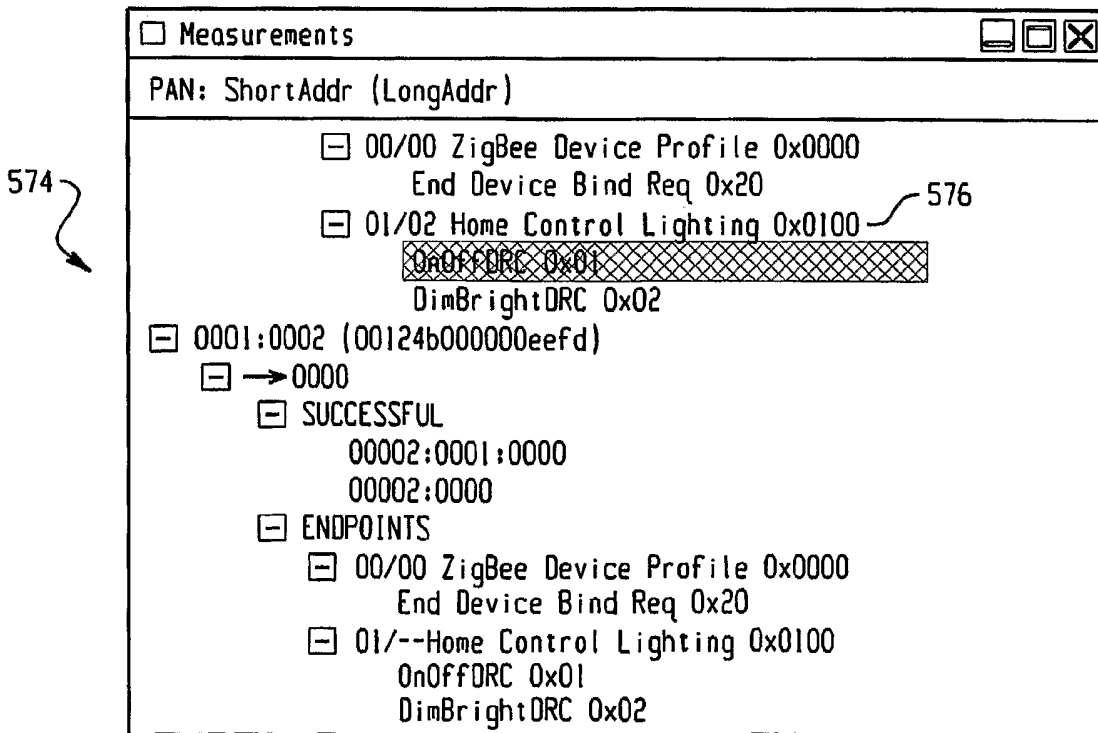


Fig. 31

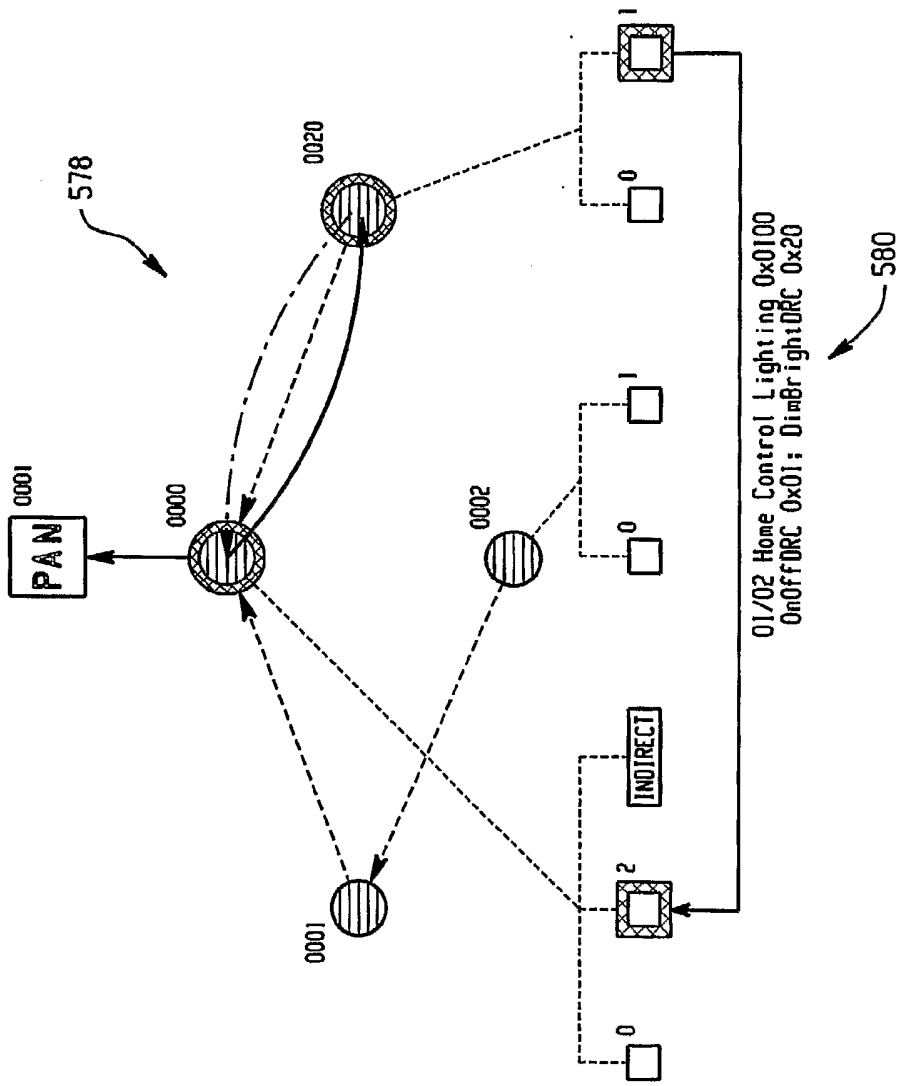


Fig. 32

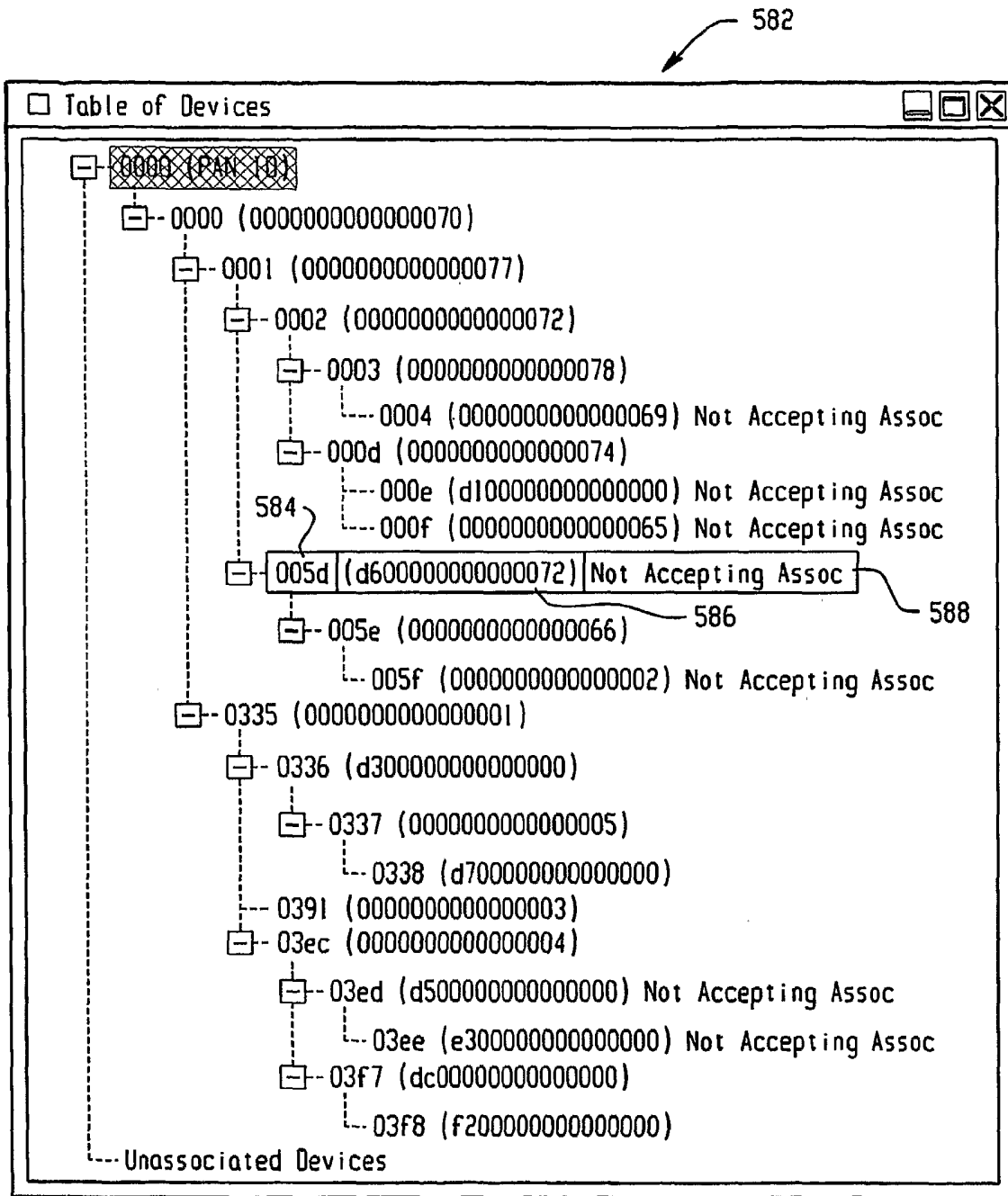


Fig. 33

590

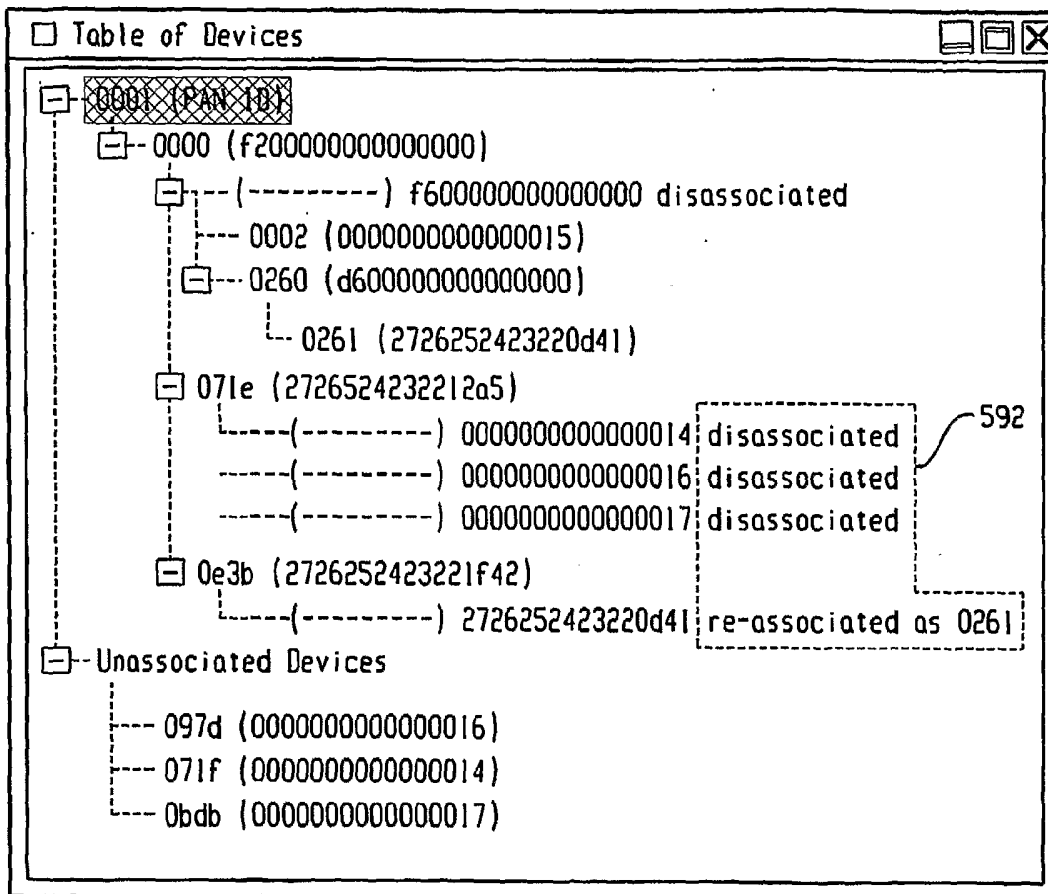


Fig. 34

Measurements		Refx Ratio	Rssi Min	Rssi Avg	Rssi Max	Uni Tx	Uni Rx	Uni Loss	Best MAC	Best NWK (Tot)
PAN:ShortAddr (LongAddr)										
0001:0000 (0012460000000001)		0.0333	255	255.0000	255	57	47	0.0000	1	1
0002						15	15			
SUCCESSFUL 0000:0001:0002 ENDPOINTS 00/00 ZigBee Device Profile 0x0000 End_Device_Bind_rgp_0xa0						606				
SUCCESSFUL 0000:0020 ENDPOINTS 00/00 ZigBee Device Profile 0x0000 End_Device_Bind_rgp_0xa0 --/01 HCL Profile 0x0100 OnEFORC 0x01						30	20		3	1
0001:0001 (0002010000000000)		0.0000	255	255.0000	255	33	28		3	1
0001:0020 (1716151413120007)		0.0000	255	255.0000	255	26	28	0.0000		
0000										
SUCCESSFUL 0020:0000 ENDPOINTS 00/00 ZigBee Device Profile 0x0000 End_Device Bind_rgp 0x20 01/02 HCL Profile 0x0100 OnEFORC 0x01 BrightDRC 0x02						17	16		3	1
0001:0002 (1716151413120007)		0.0000	255	255.0000	255	15	15	0.0000		
0000										
SUCCESSFUL 0002:0001:0000 0002:0000 ENDPOINTS										

Fig. 35

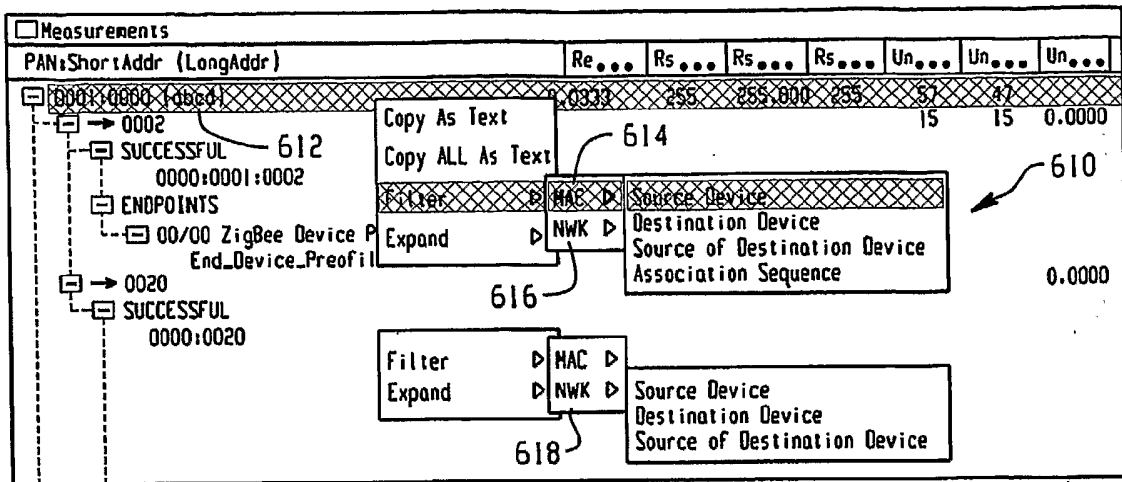


Fig. 36

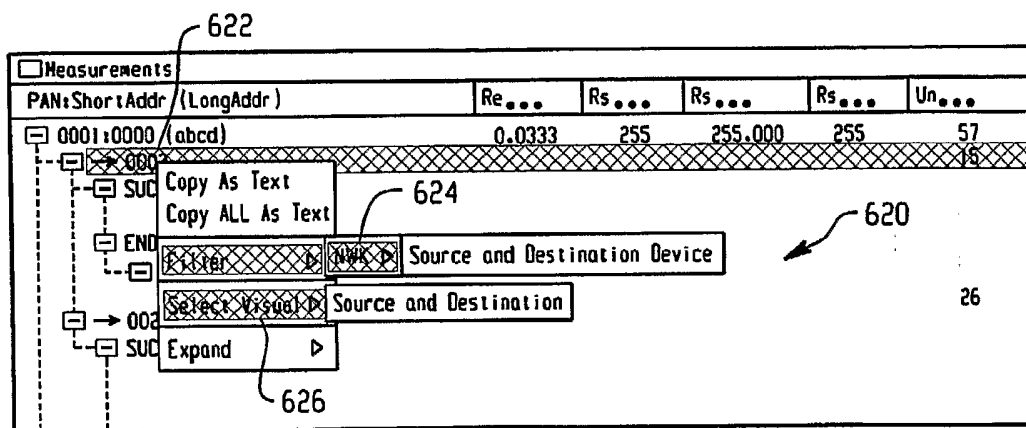


Fig. 37

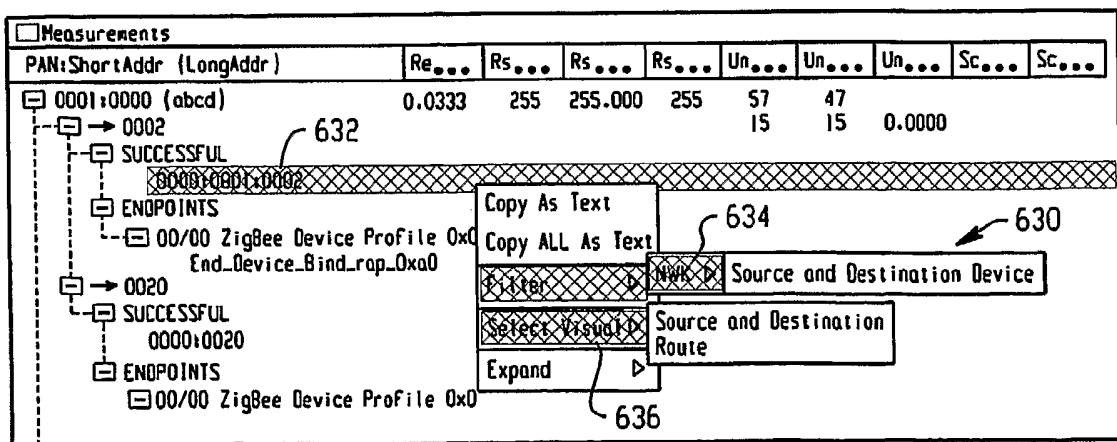


Fig. 38

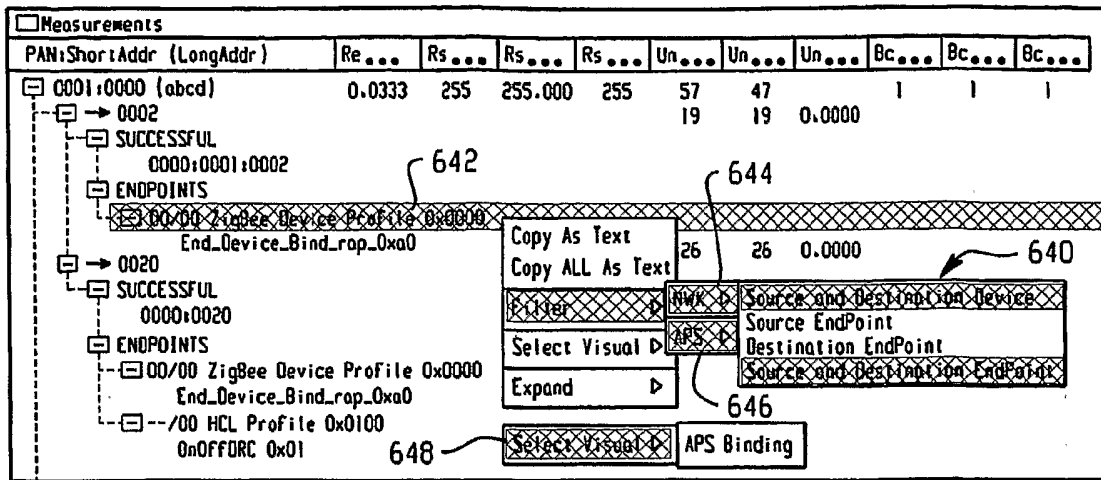


Fig. 39

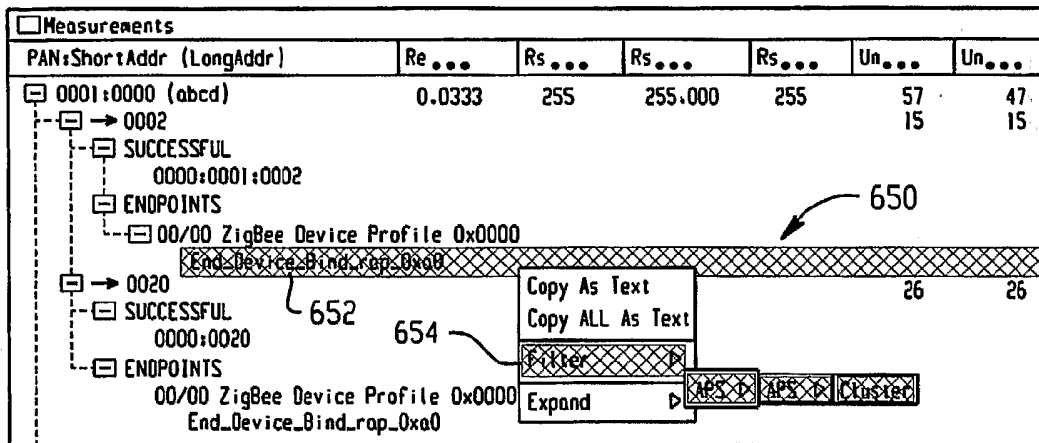


Fig. 40

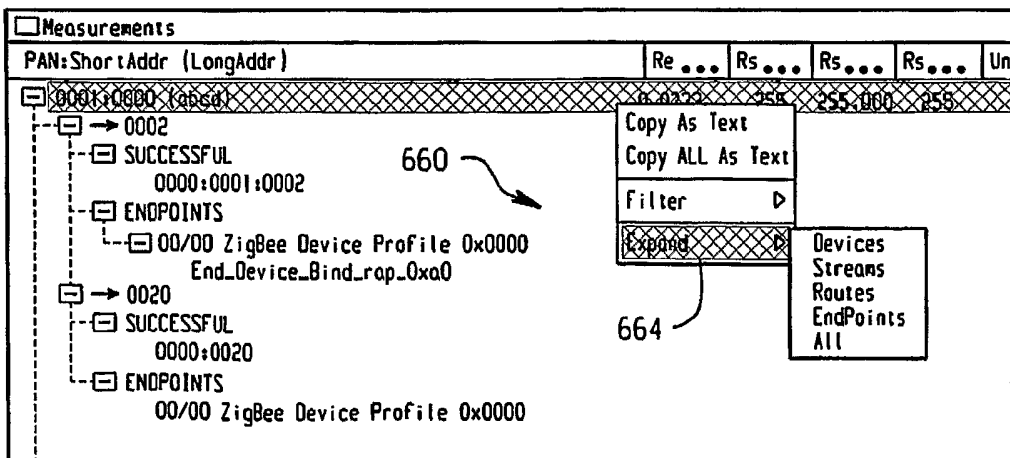


Fig. 41

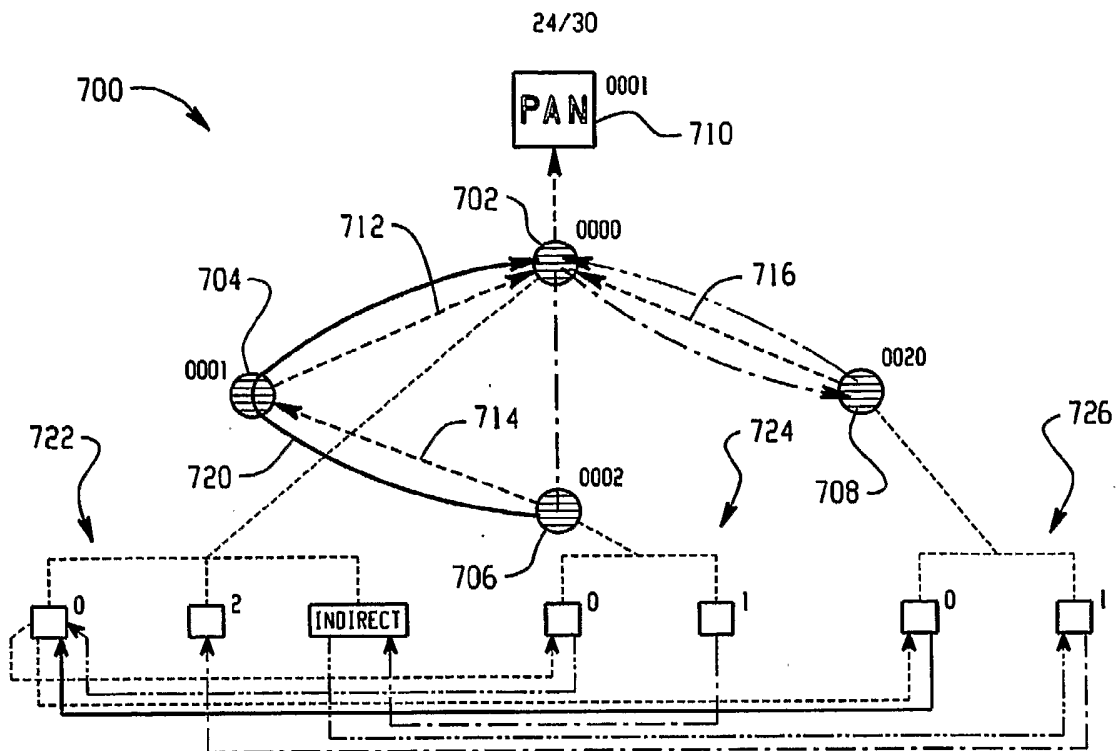


Fig. 42

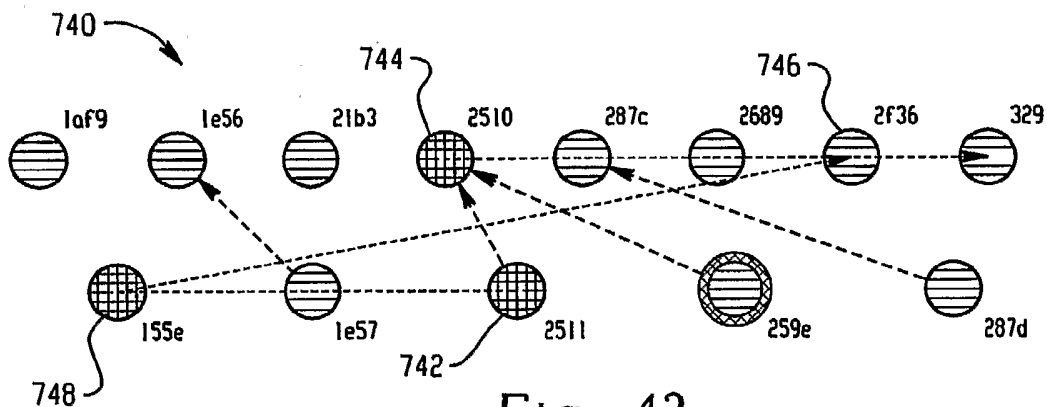


Fig. 43

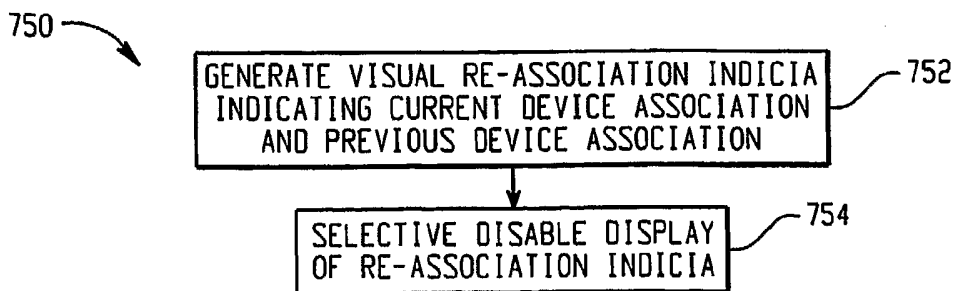


Fig. 44

25/30

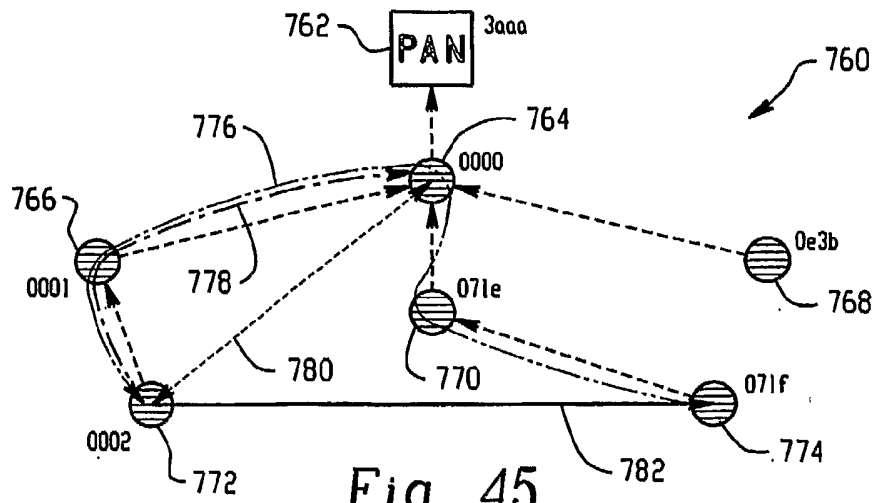


Fig. 45

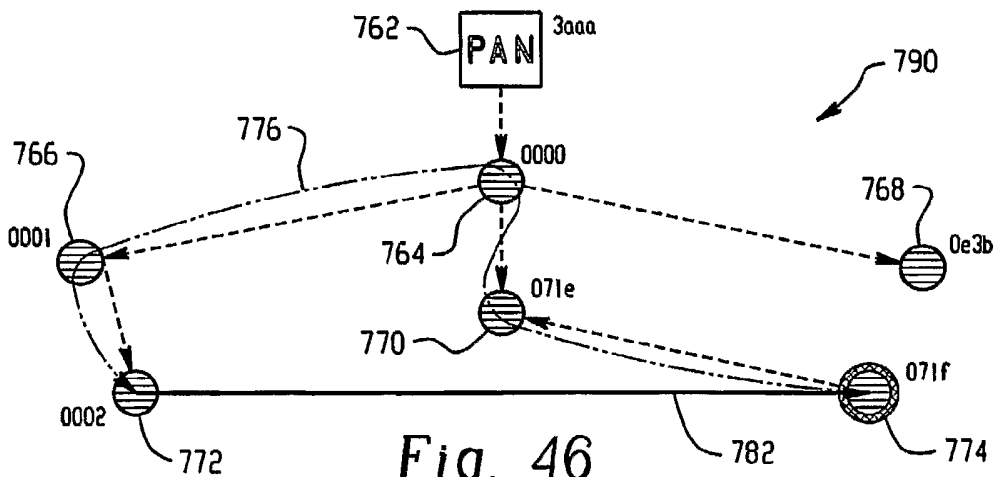


Fig. 46

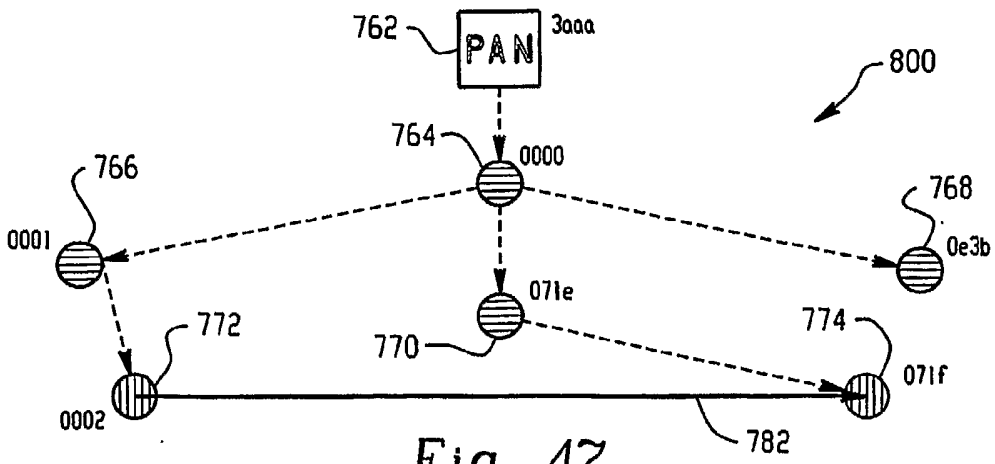


Fig. 47

26/30

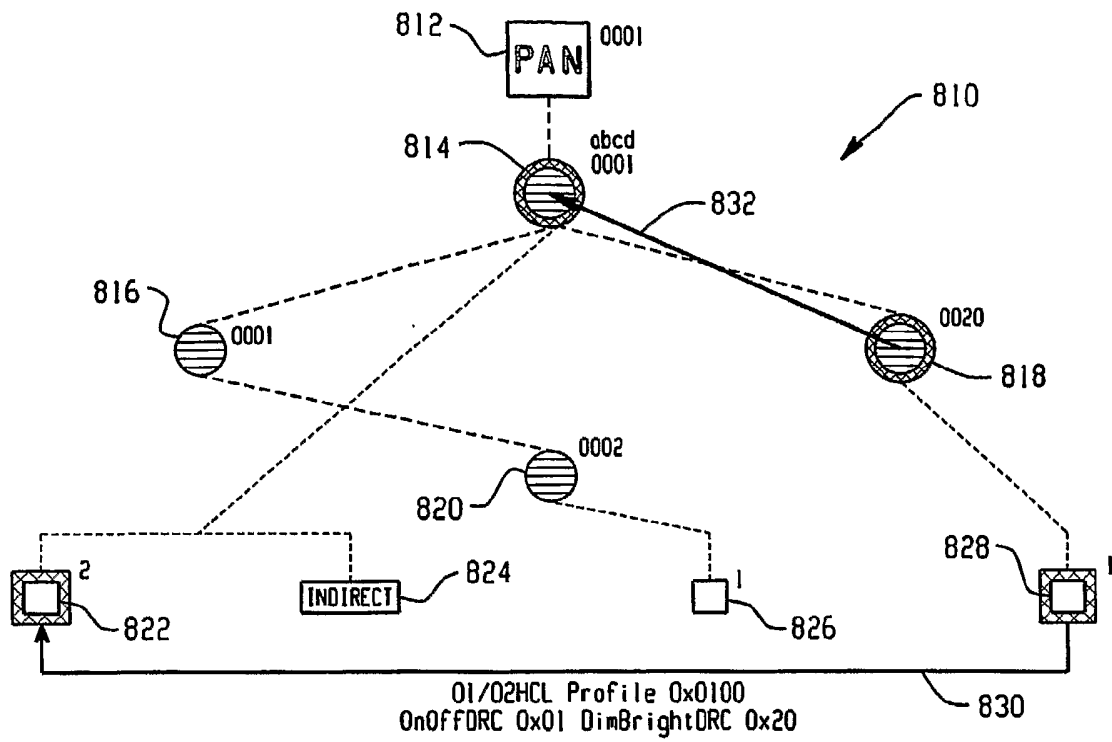


Fig. 48

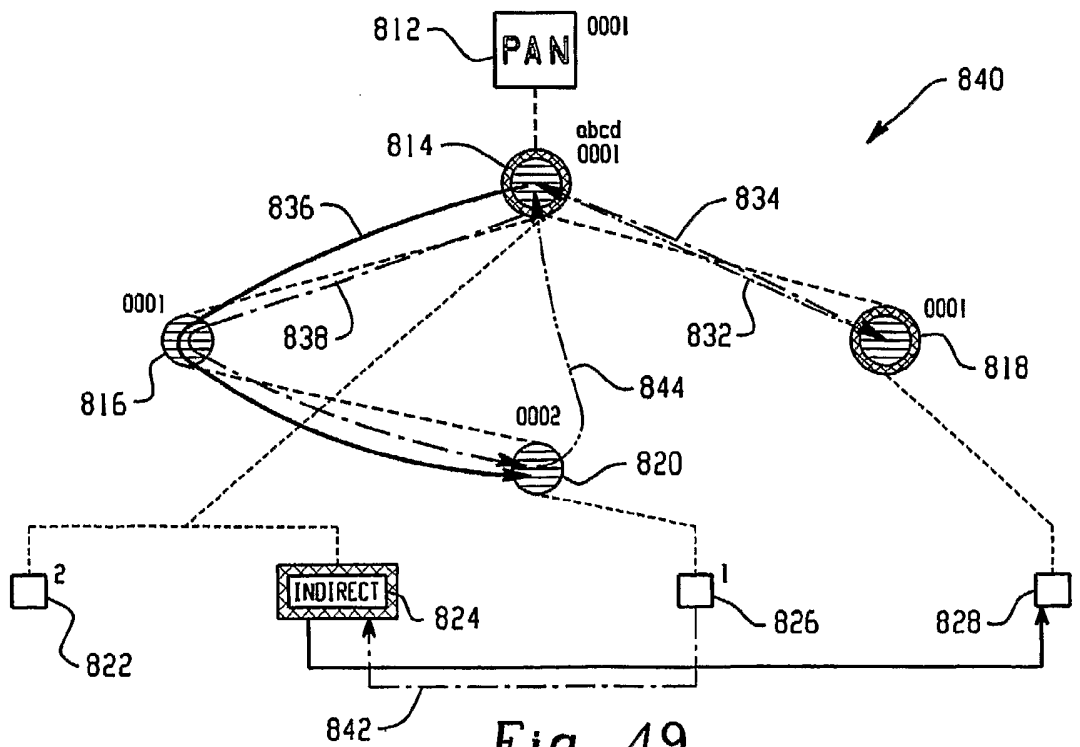


Fig. 49

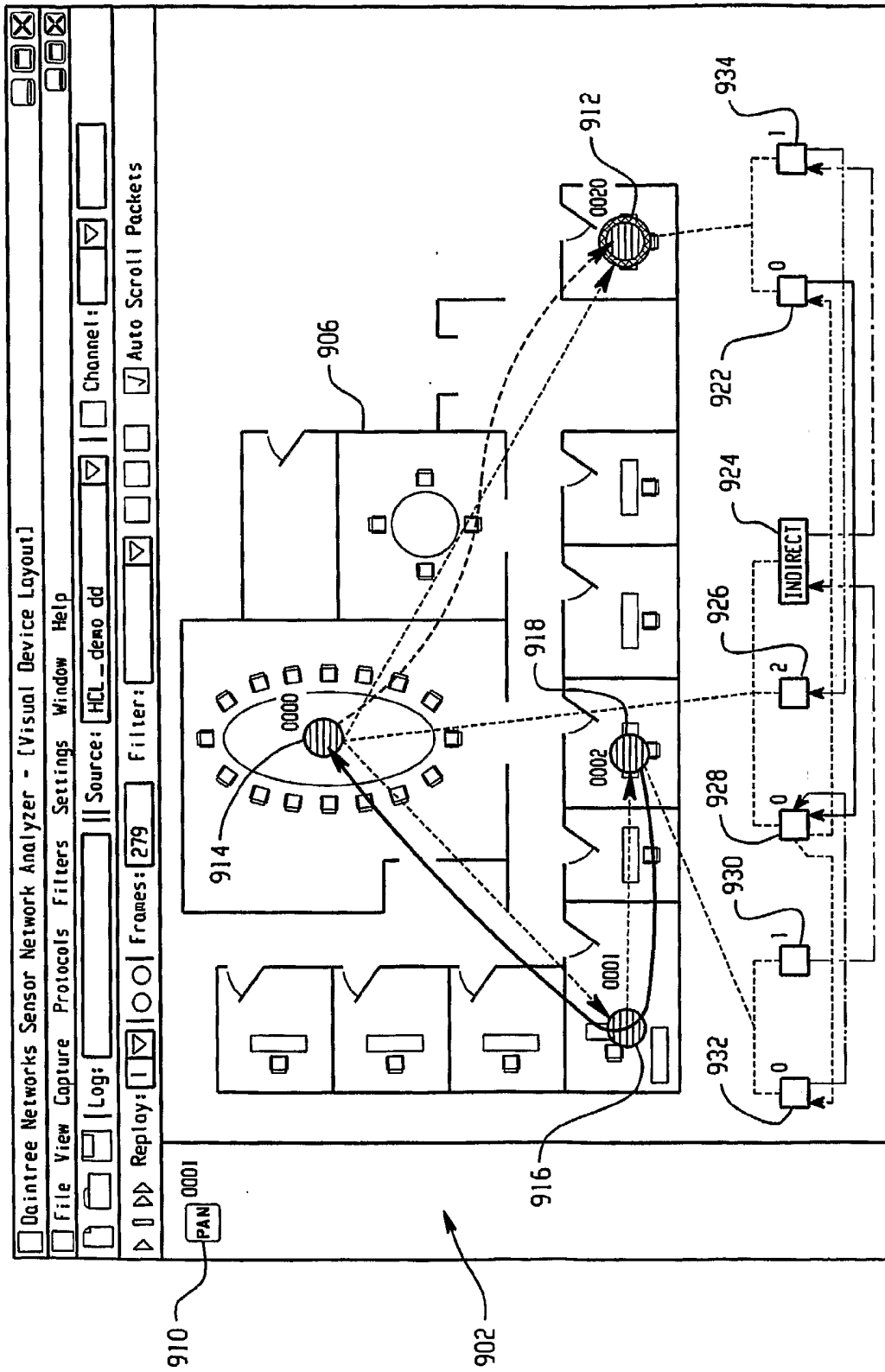


Fig. 51

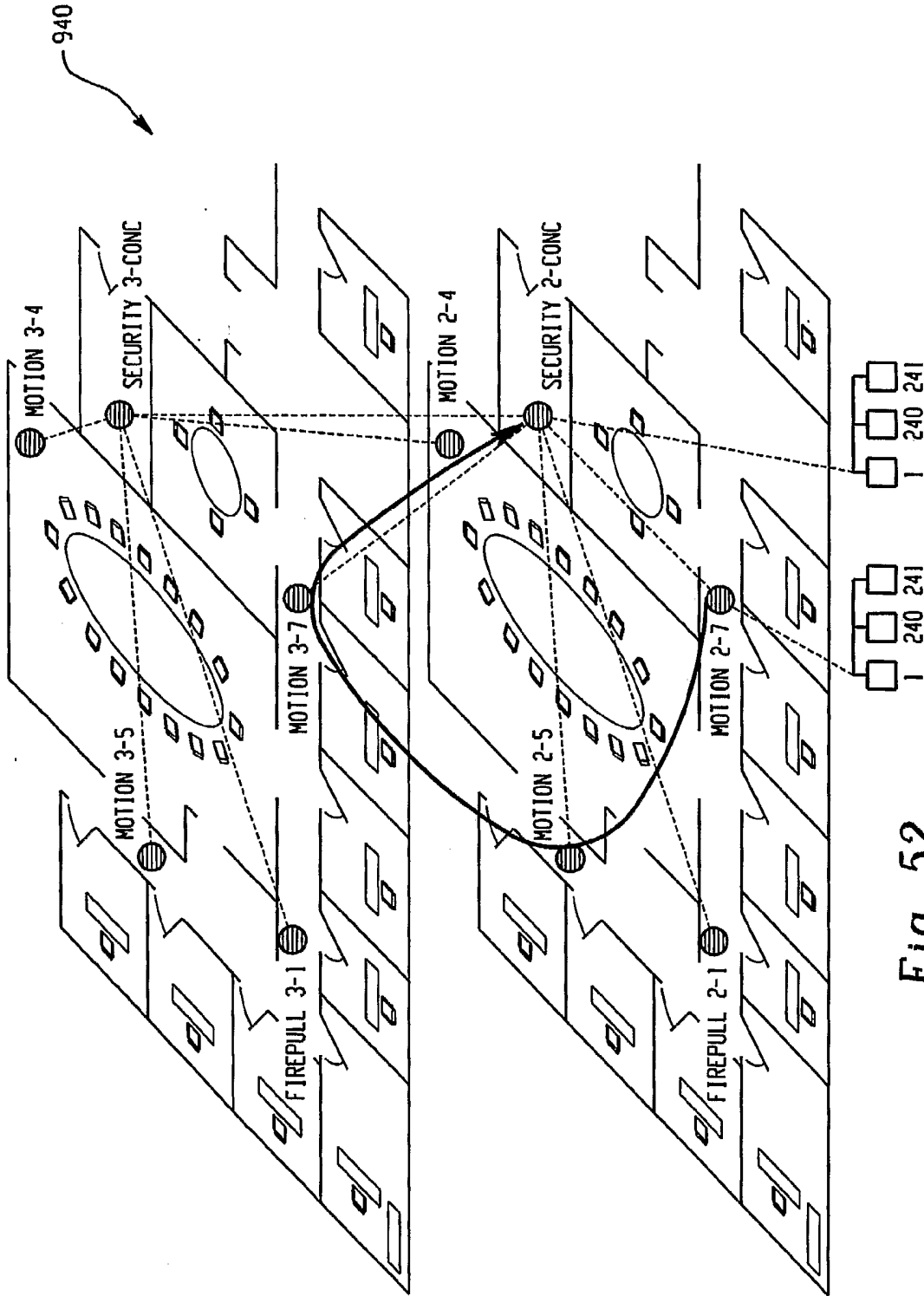


Fig. 52

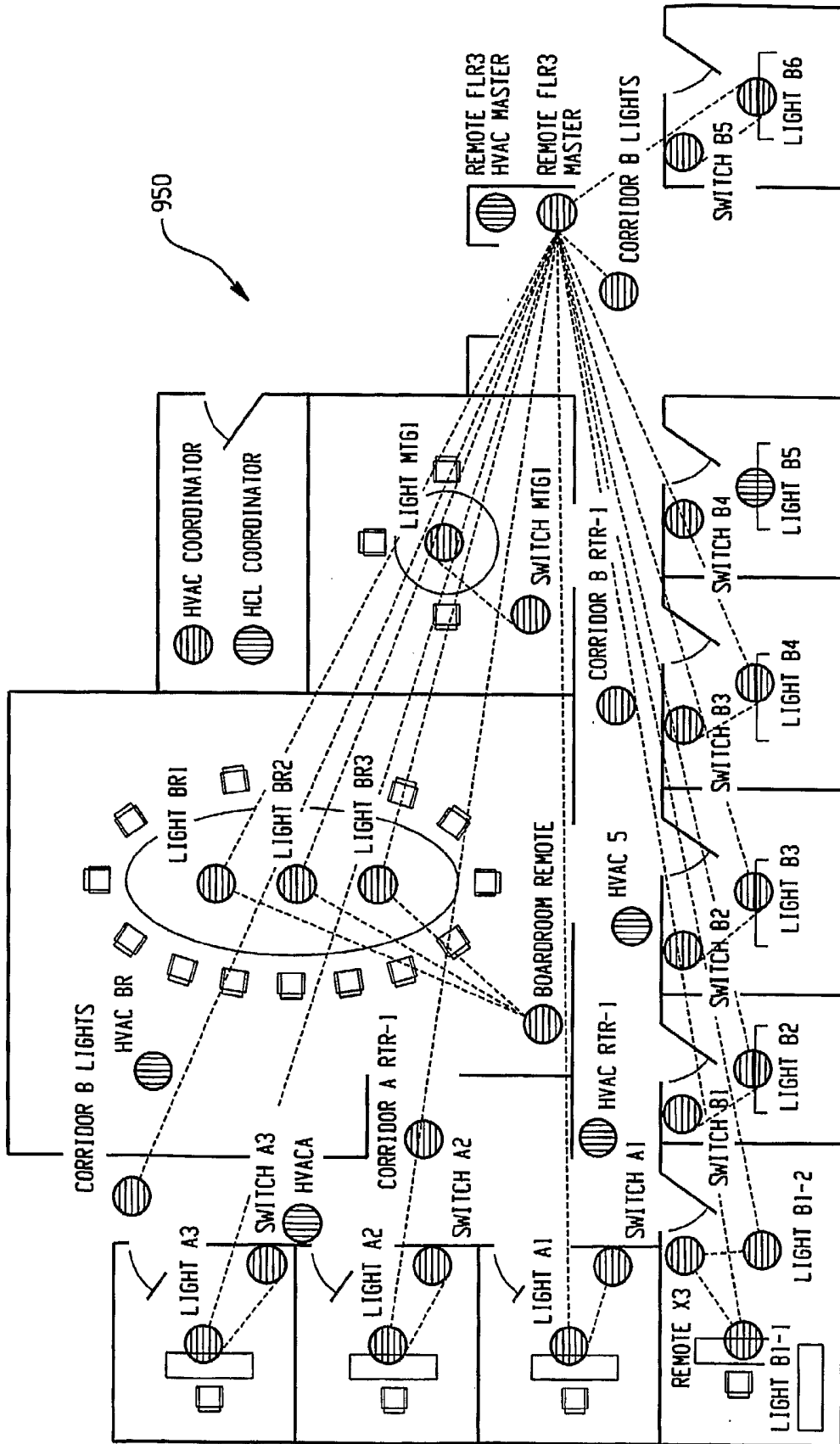


Fig. 53