



(19) **United States**
(12) **Patent Application Publication**
VEILLETTE

(10) **Pub. No.: US 2009/0135836 A1**
(43) **Pub. Date: May 28, 2009**

(54) **COLLECTOR DEVICE AND SYSTEM
UTILIZING STANDARDIZED UTILITY
METERING PROTOCOL**

(76) Inventor: **Michel VEILLETTE**, Waterloo
(CA)

Correspondence Address:
King and Spalding LLP (Trilliant)
Trilliant Customer Number
1700 Pennsylvania Avenue, NW, Suite 200
Washington, DC 20006 (US)

(21) Appl. No.: **12/275,242**

(22) Filed: **Nov. 21, 2008**

Related U.S. Application Data

(60) Provisional application No. 60/989,957, filed on Nov. 25, 2007, provisional application No. 60/989,967, filed on Nov. 25, 2007, provisional application No. 60/989,958, filed on Nov. 25, 2007, provisional application No. 60/989,964, filed on Nov. 25, 2007, provisional application No. 60/989,950, filed on Nov. 25, 2007, provisional application No. 60/989,953, filed on Nov. 25, 2007, provisional application No. 60/989,975, filed on Nov. 25, 2007, provisional application No. 60/989,959, filed on Nov. 25, 2007, provisional application No. 60/989,961, filed on Nov. 25, 2007, provisional application No. 60/989,962, filed on Nov. 25, 2007, provisional application No. 60/989,951, filed on Nov. 25, 2007, provisional application No. 60/989,955, filed on Nov. 25, 2007, provisional appli-

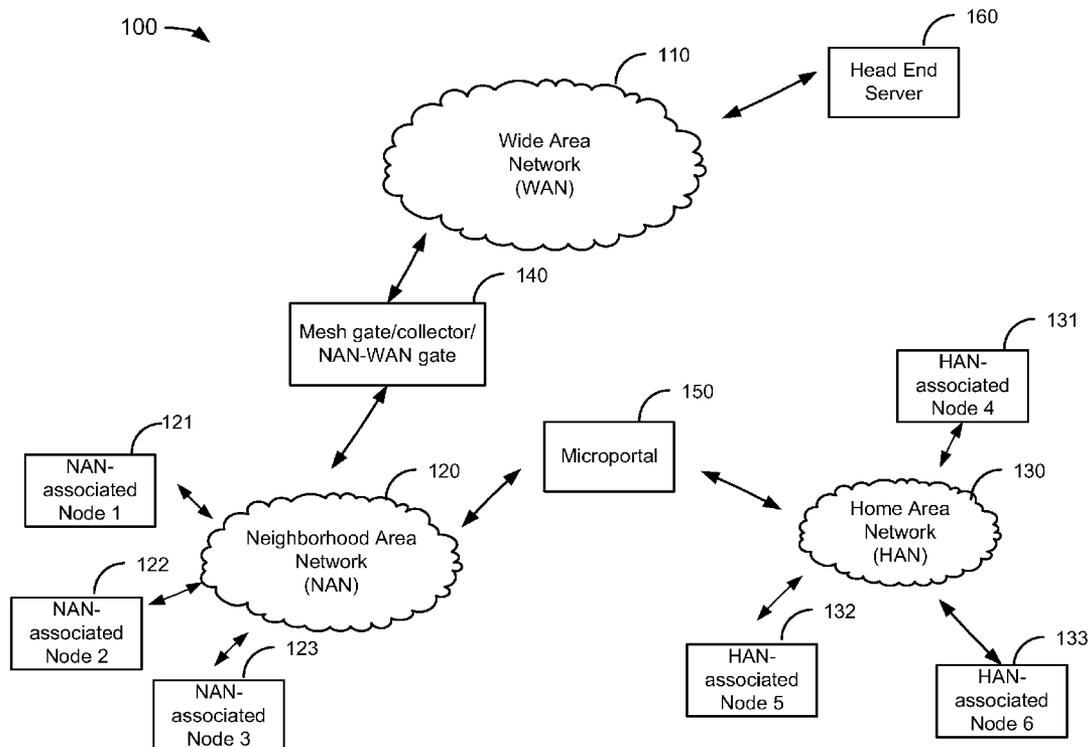
cation No. 60/989,952, filed on Nov. 25, 2007, provisional application No. 60/989,954, filed on Nov. 25, 2007, provisional application No. 60/992,312, filed on Dec. 4, 2007, provisional application No. 60/992,313, filed on Dec. 4, 2007, provisional application No. 60/992,315, filed on Dec. 4, 2007, provisional application No. 61/025,279, filed on Jan. 31, 2008, provisional application No. 61/025,270, filed on Jan. 31, 2008, provisional application No. 61/025,276, filed on Jan. 31, 2008, provisional application No. 61/025,282, filed on Jan. 31, 2008, provisional application No. 61/025,271, filed on Jan. 31, 2008, provisional application No. 61/025,287, filed on Jan. 31, 2008, provisional application No. 61/025,278, filed on Jan. 31, 2008, provisional application No. 61/025,273, filed on Jan. 31, 2008, provisional application No. 61/025,277, filed on Jan. 31, 2008, provisional application No. 61/094,116, filed on Sep. 4, 2008.

Publication Classification

(51) **Int. Cl.**
H04L 12/56 (2006.01)
(52) **U.S. Cl.** **370/400**

(57) **ABSTRACT**

A device-agnostic collector is capable of accepting information pushed from a node for aggregation and subsequent transmission to the head end server. As a receiver of pushed information, the device-agnostic collector does not require information about the type of meter at the node or the type of data structures transmitted by the node.



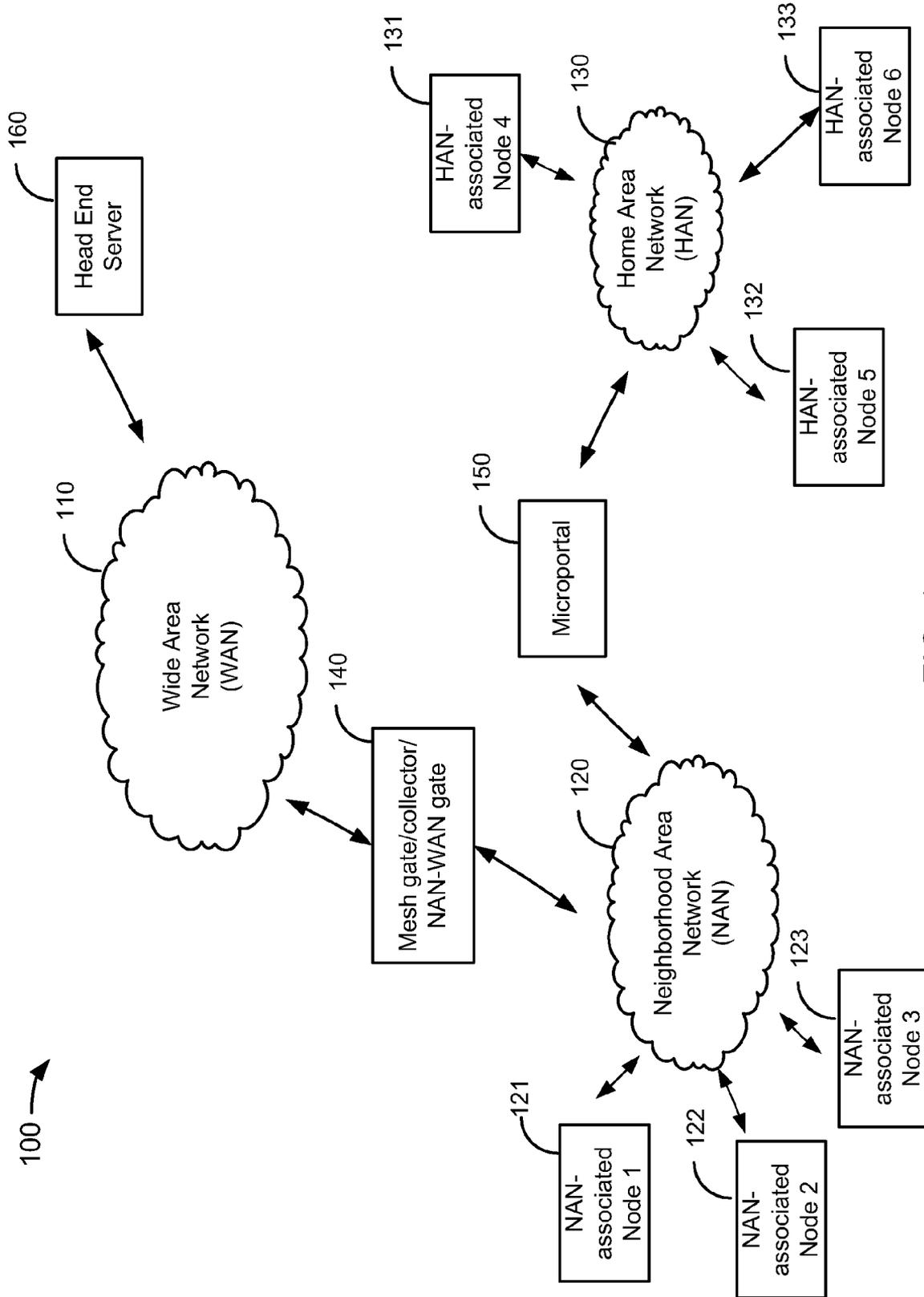


FIG. 1

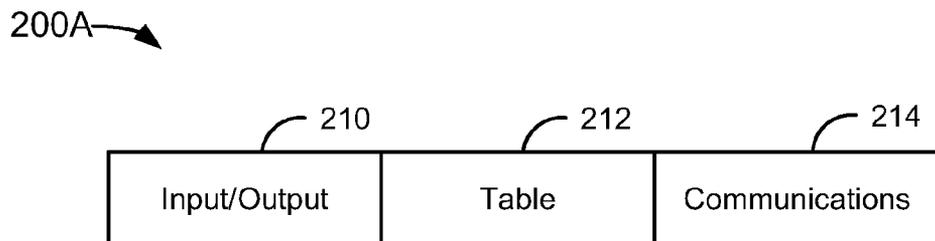


FIG. 2A

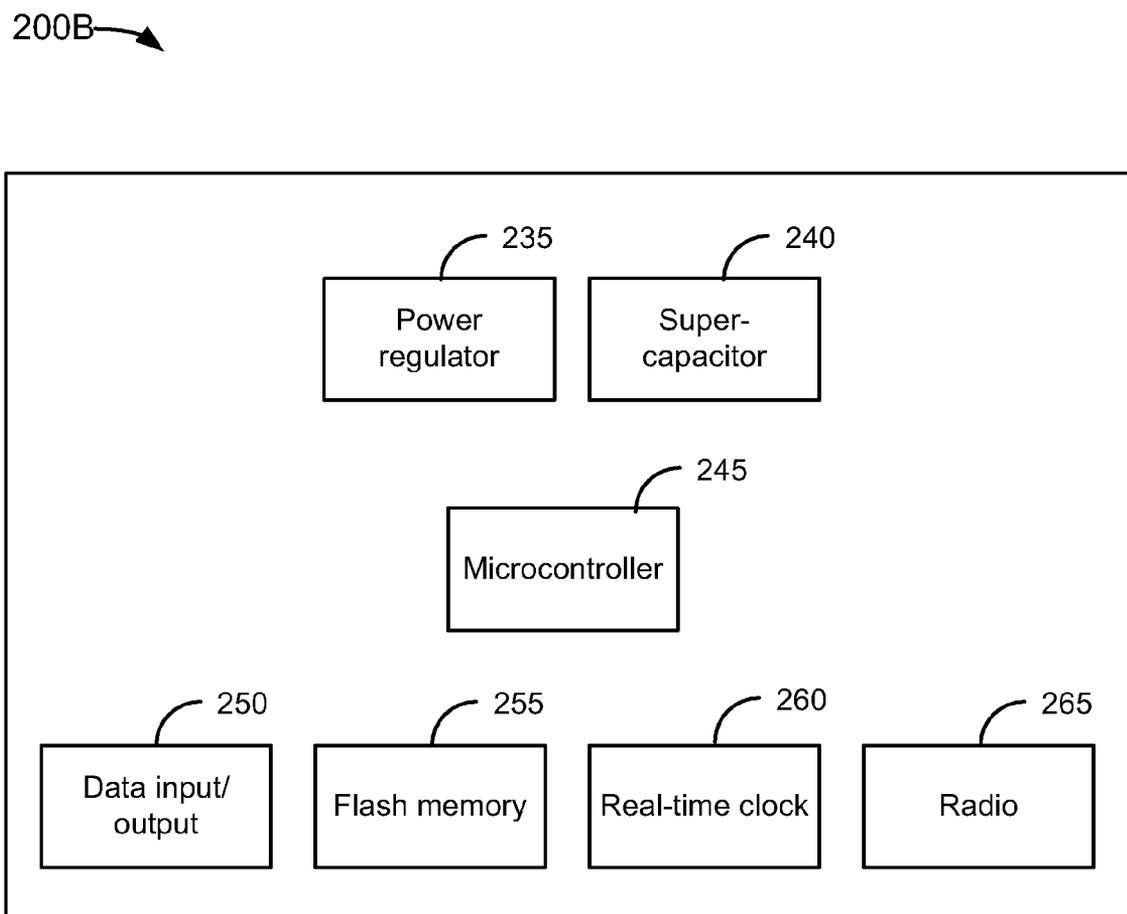


FIG. 2B

300 →

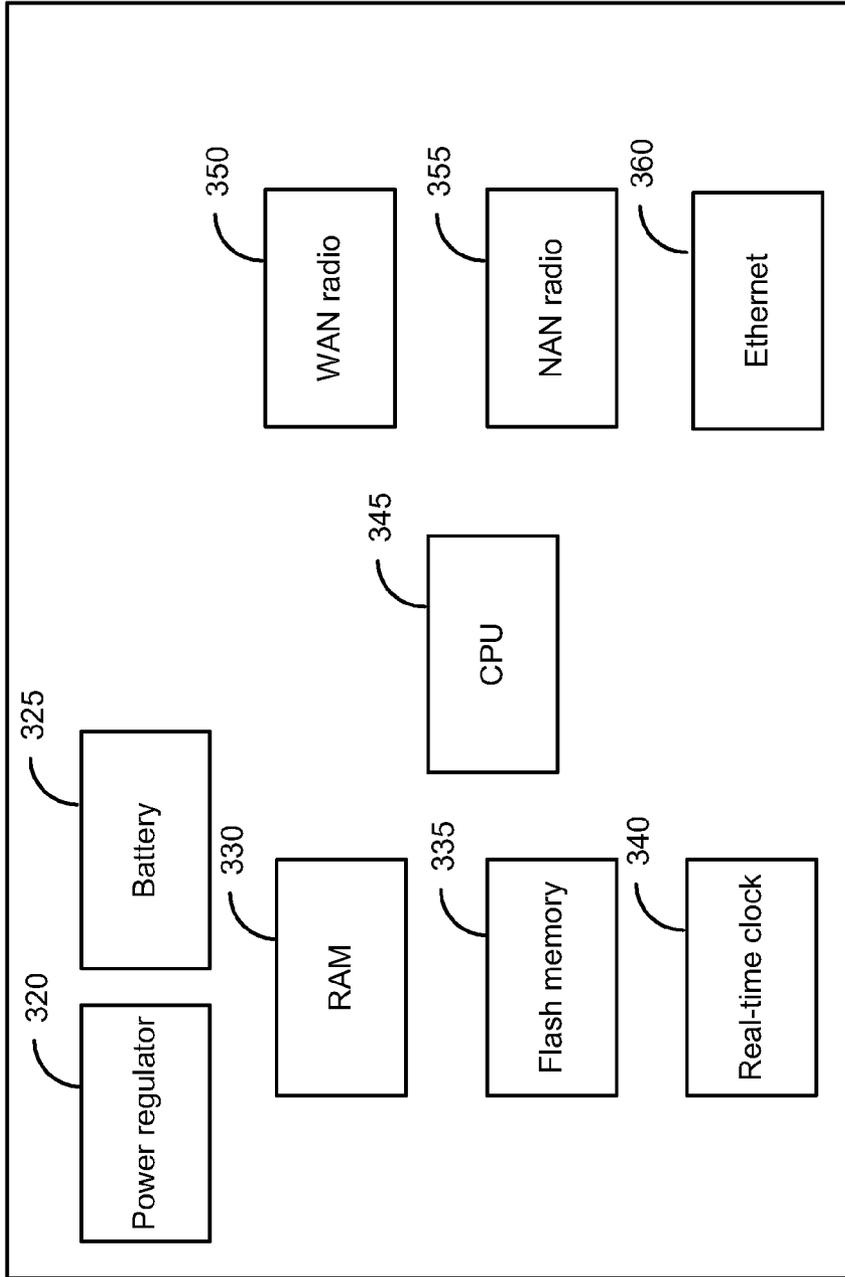


FIG. 3

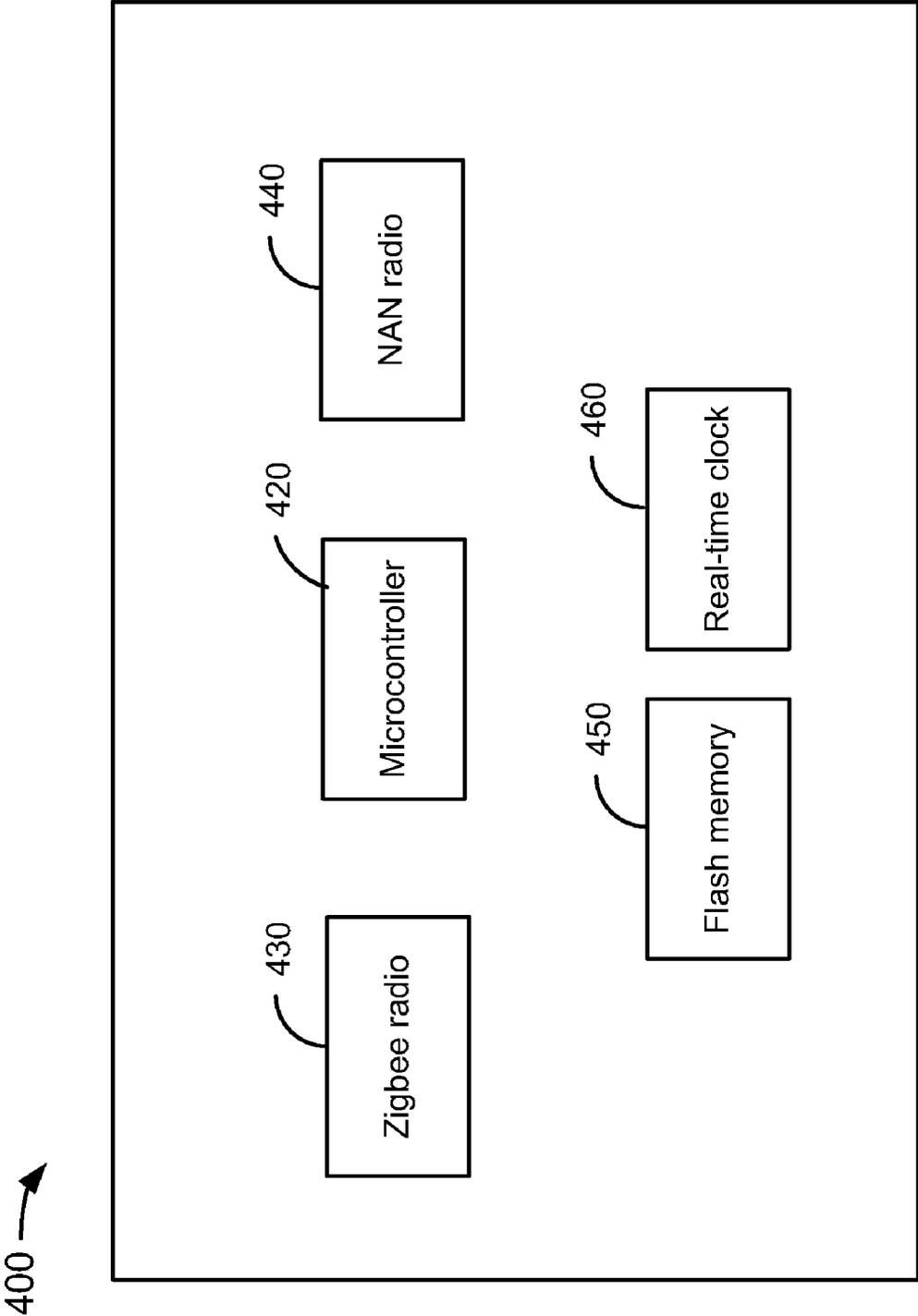


FIG. 4

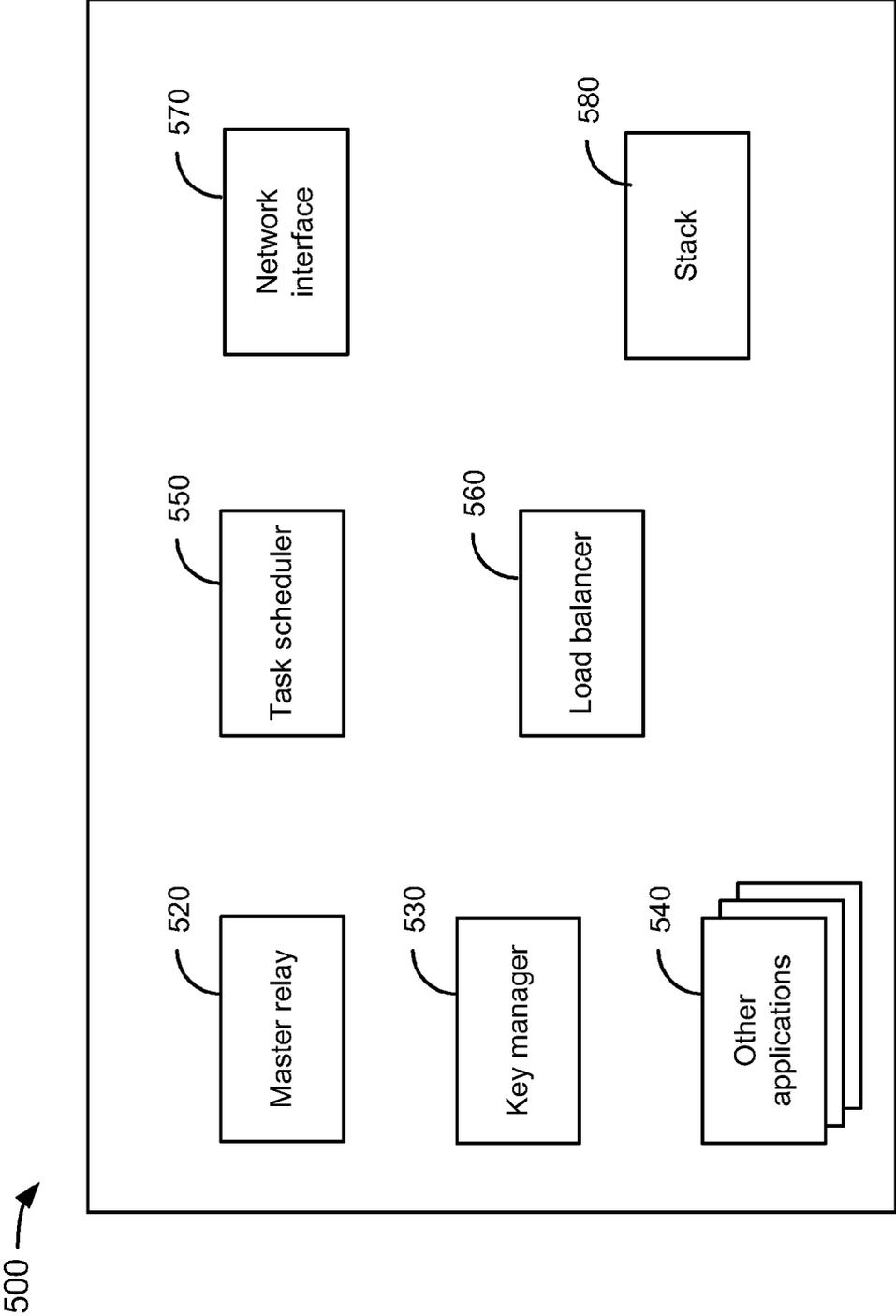


FIG. 5

600A →

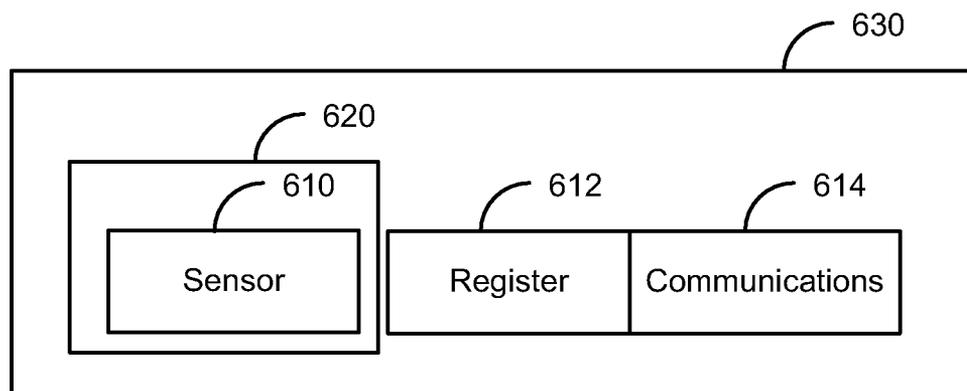


FIG. 6A

600B →

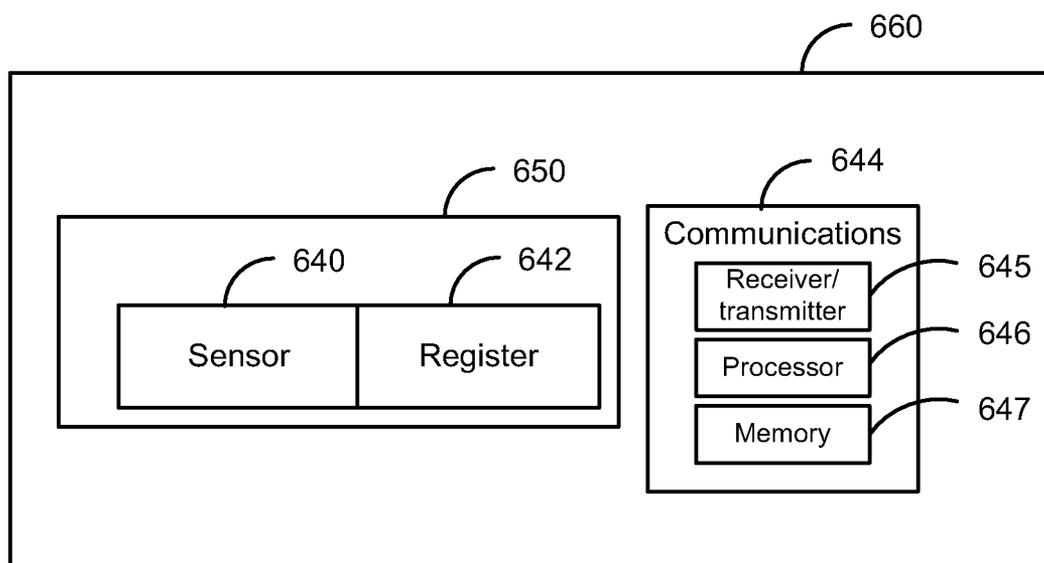


FIG. 6B

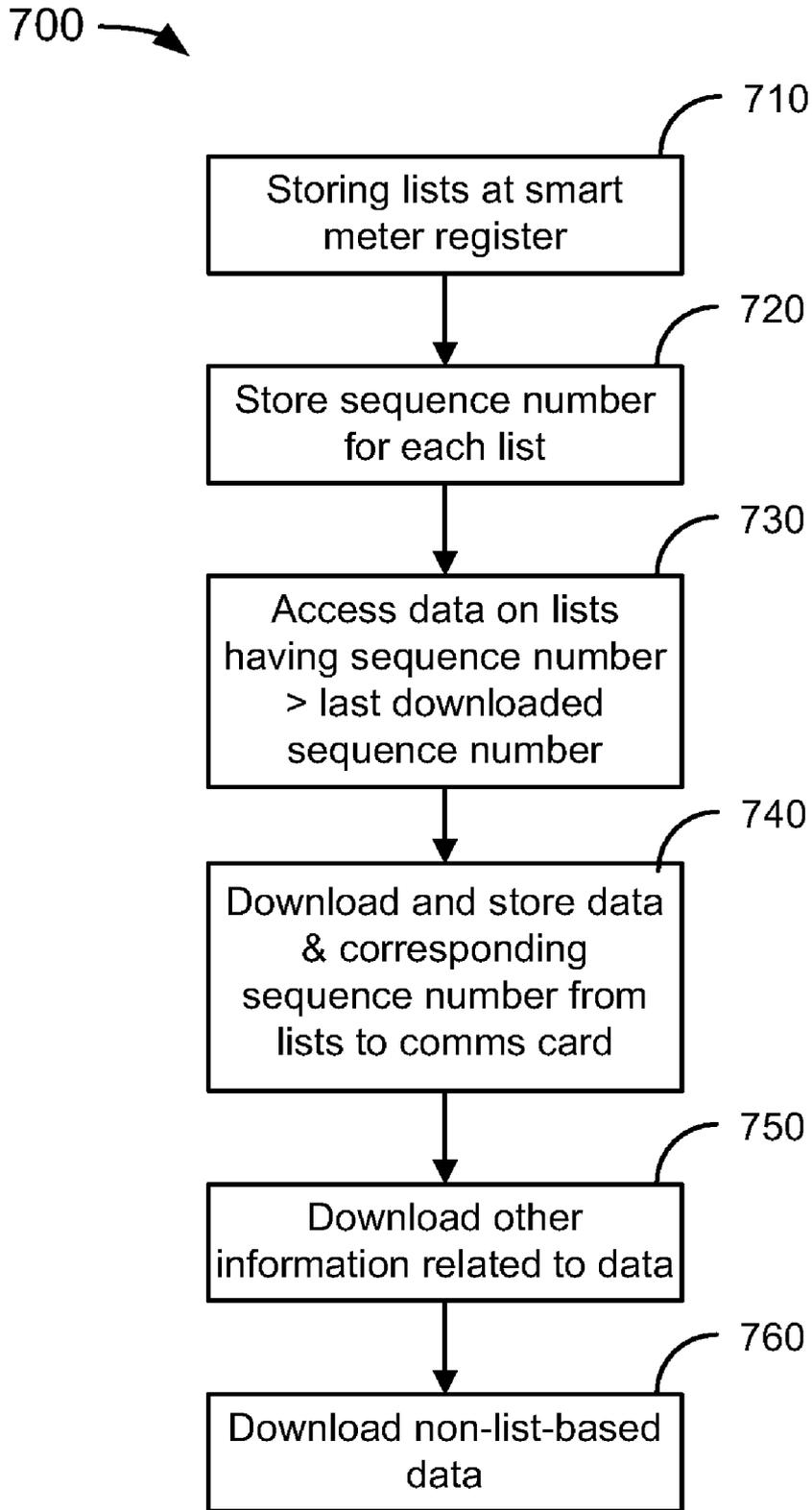


FIG. 7

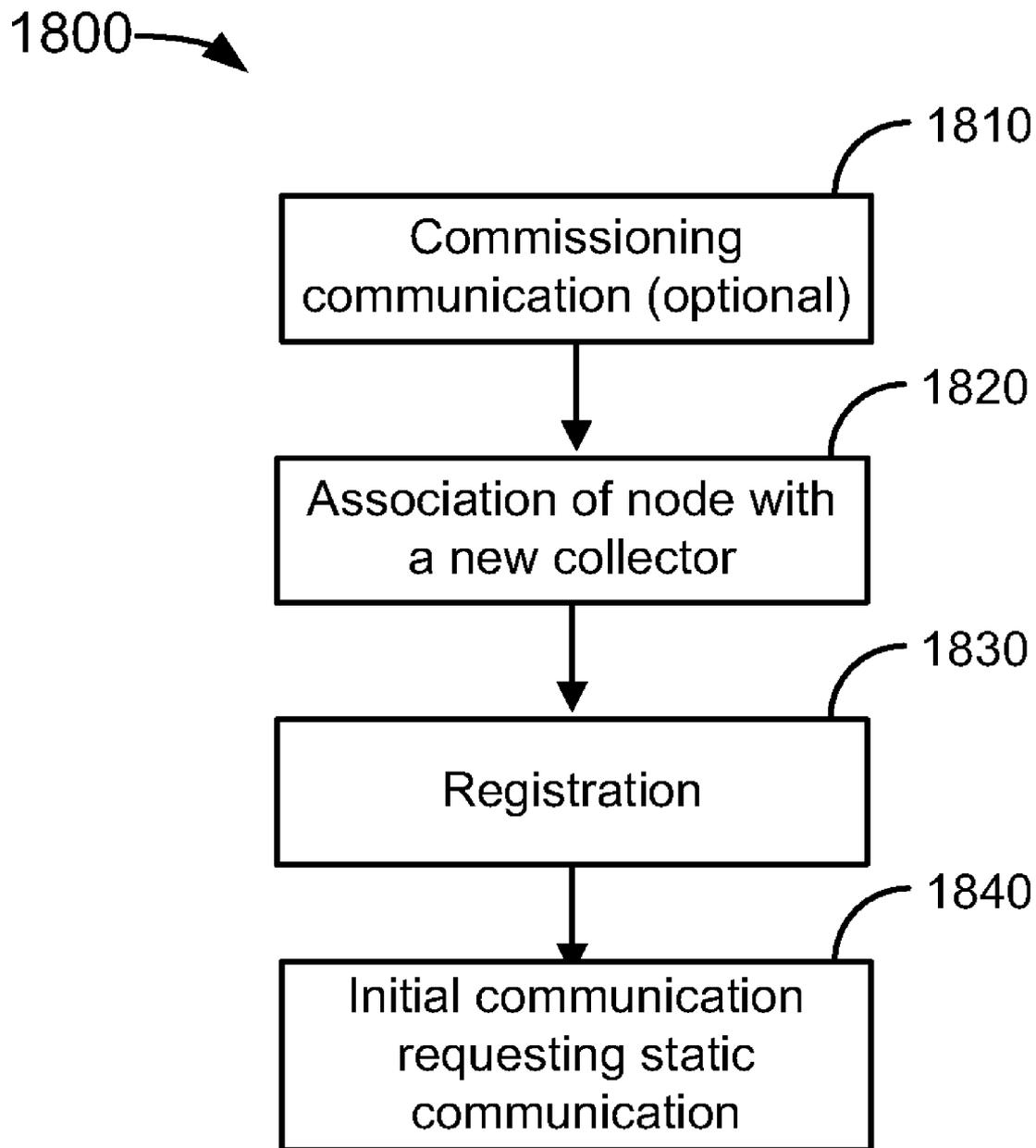


FIG. 8

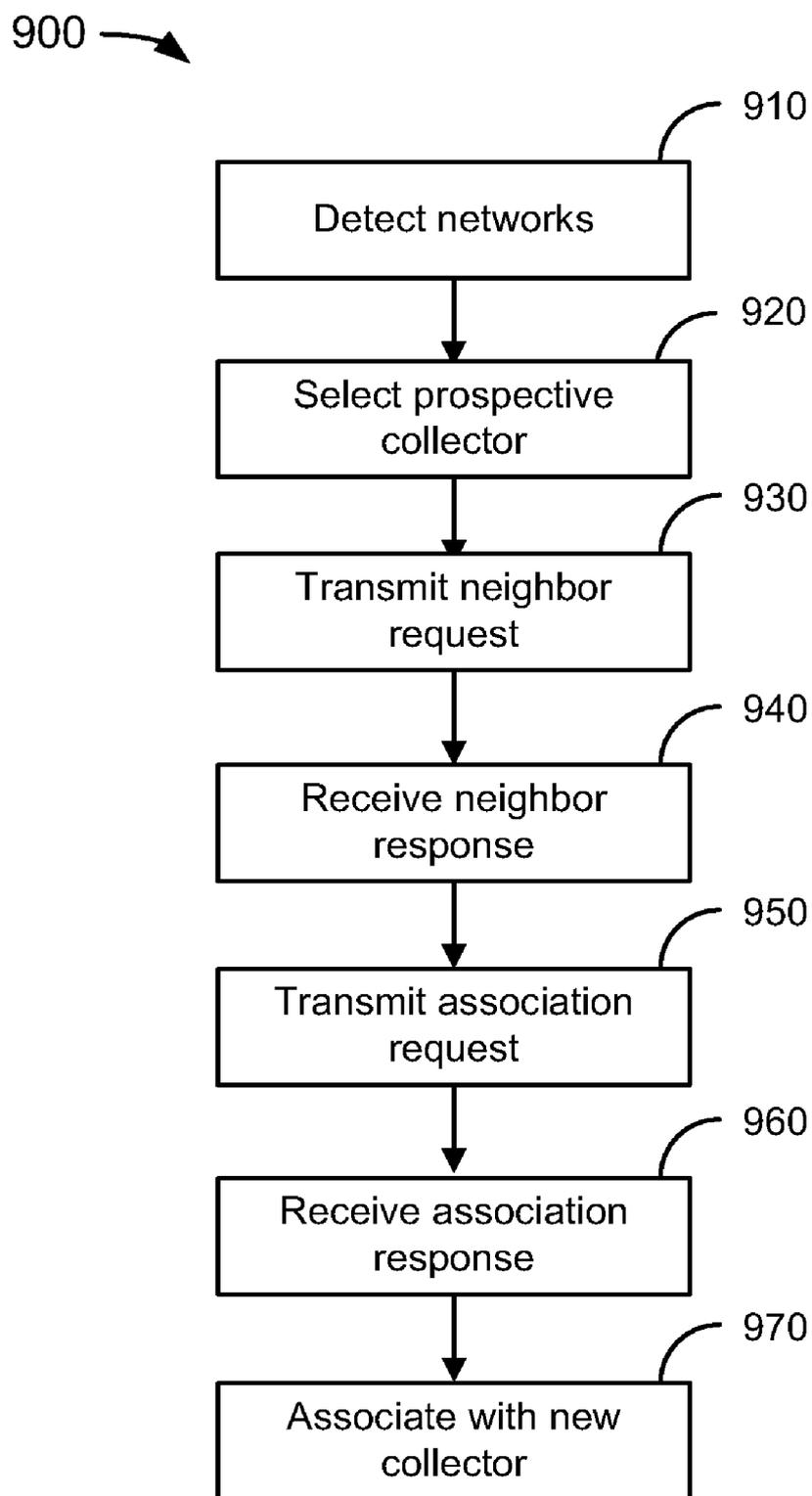


FIG. 9

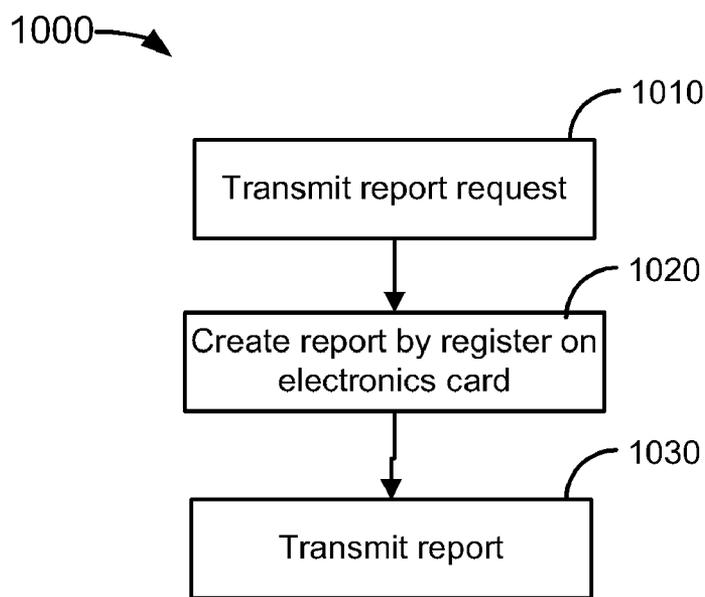


FIG. 10

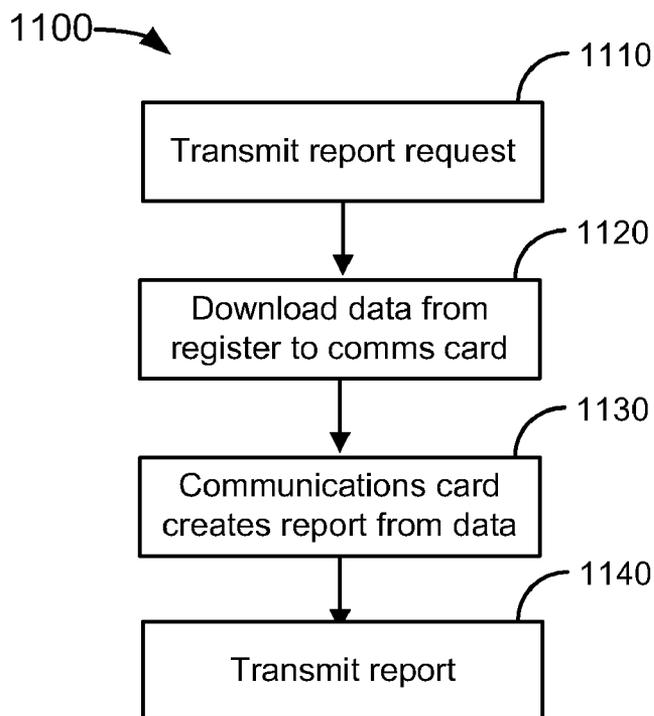


FIG. 11

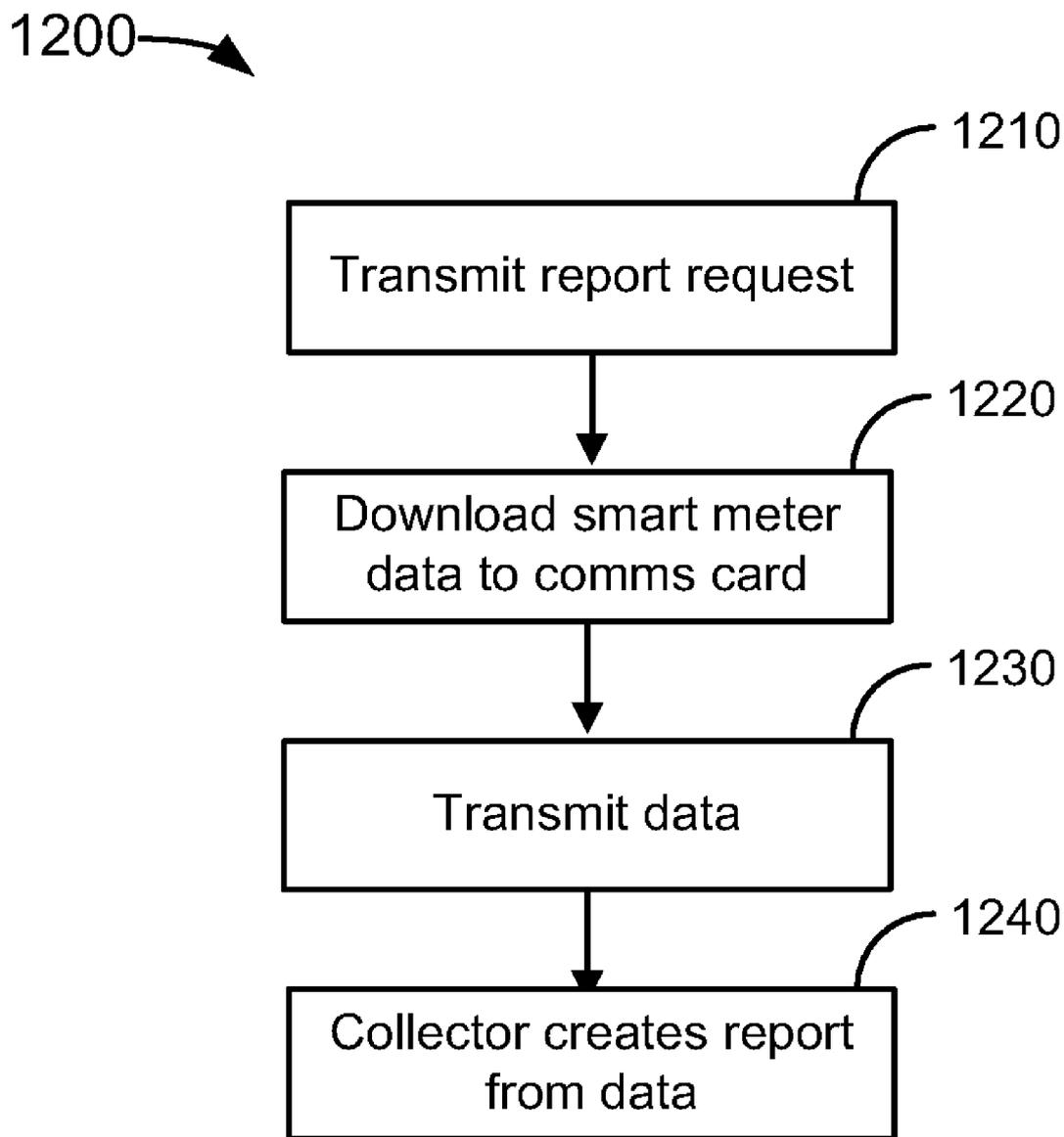


FIG. 12

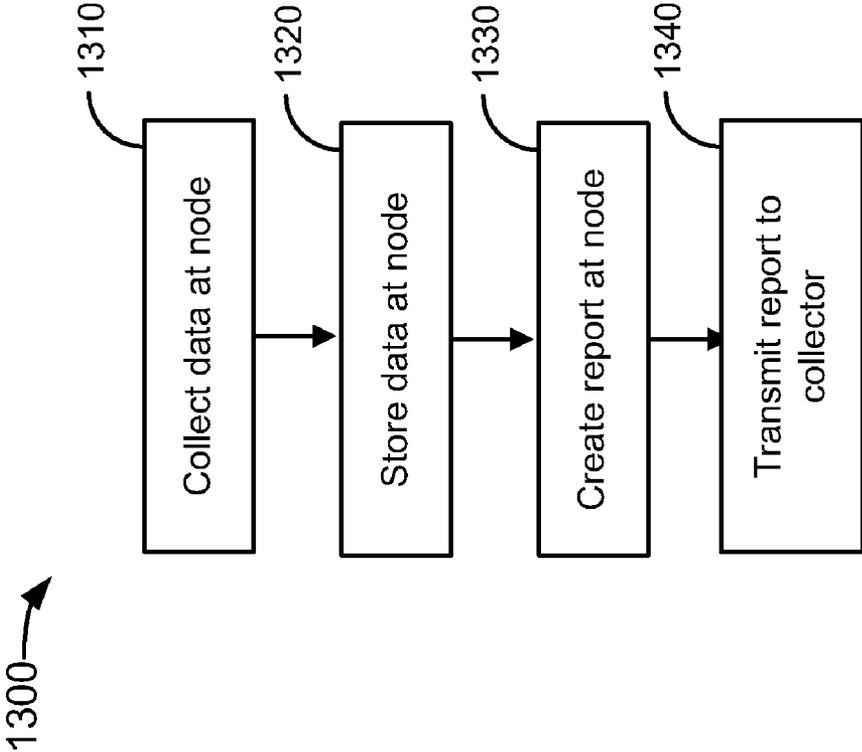


FIG. 13

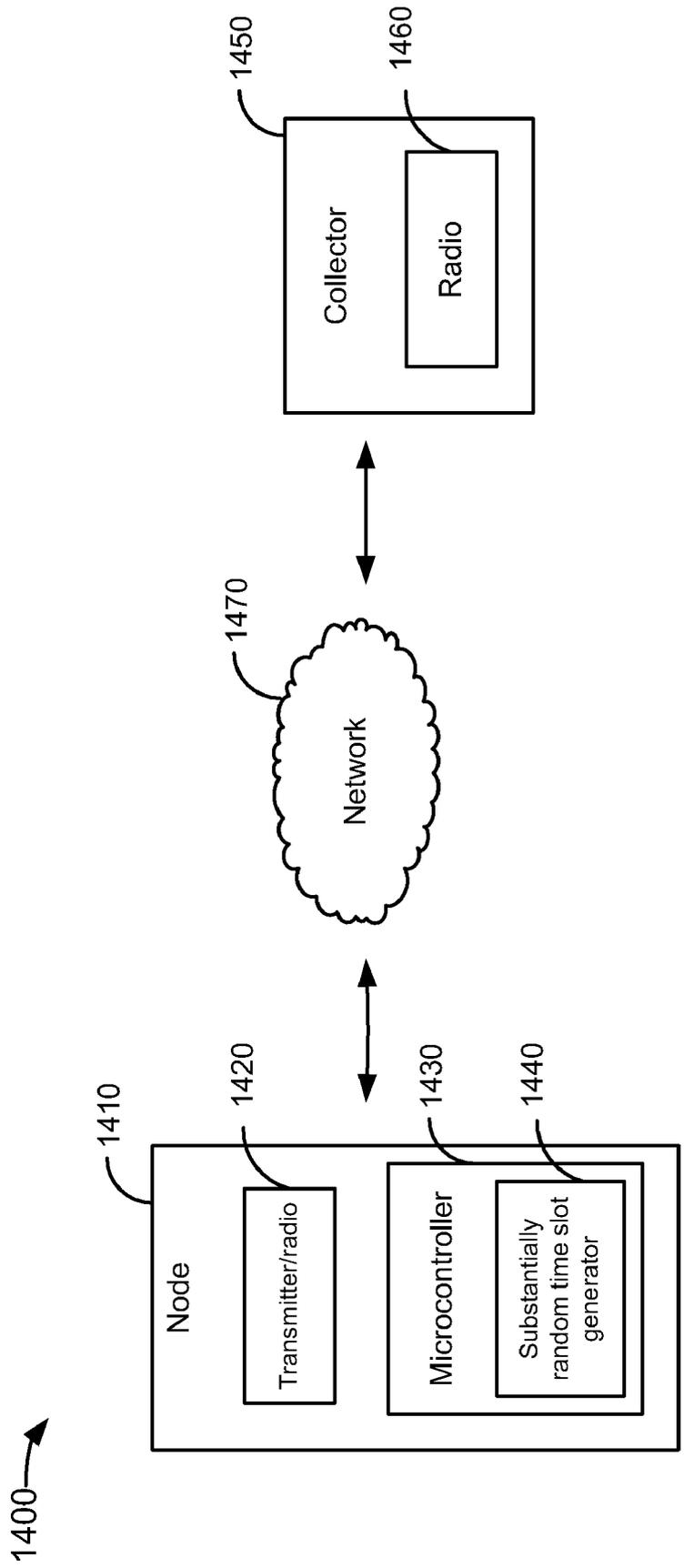


FIG. 14

1500 →

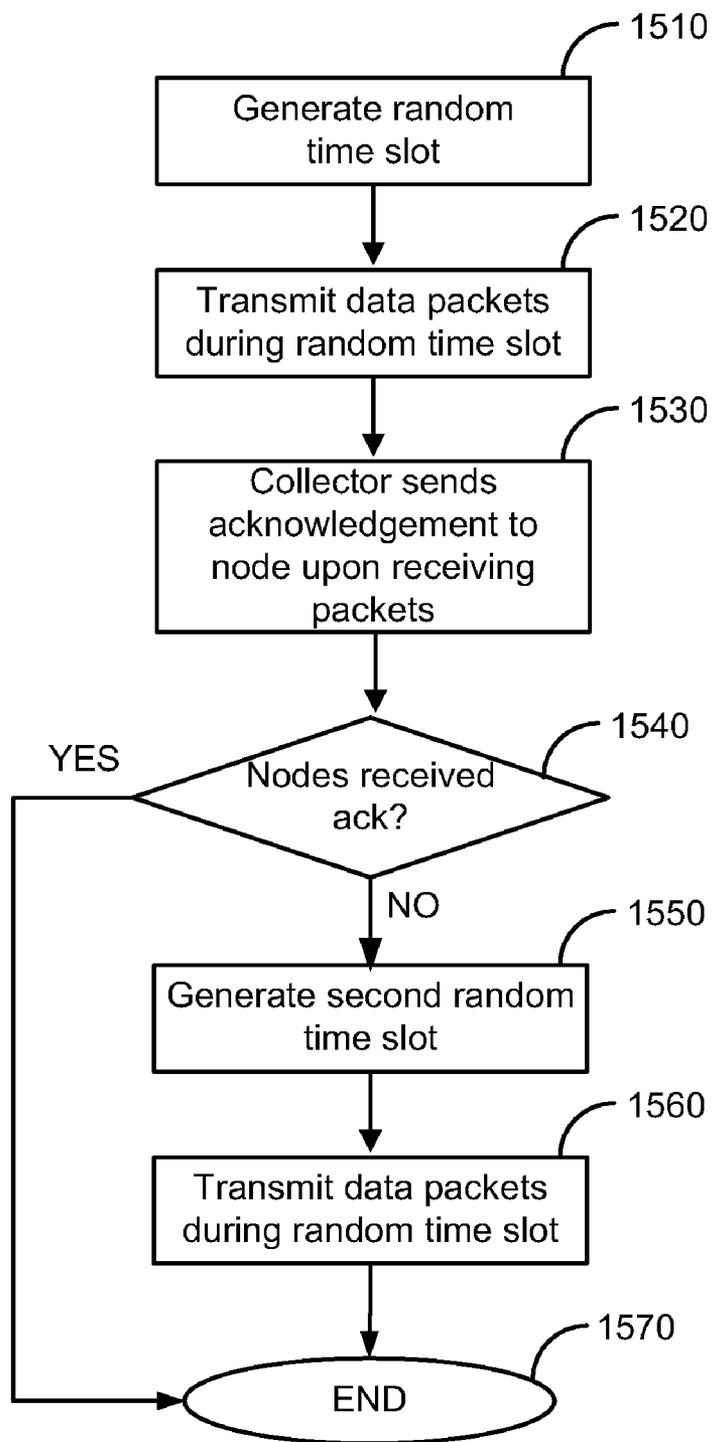


FIG. 15

1600 →

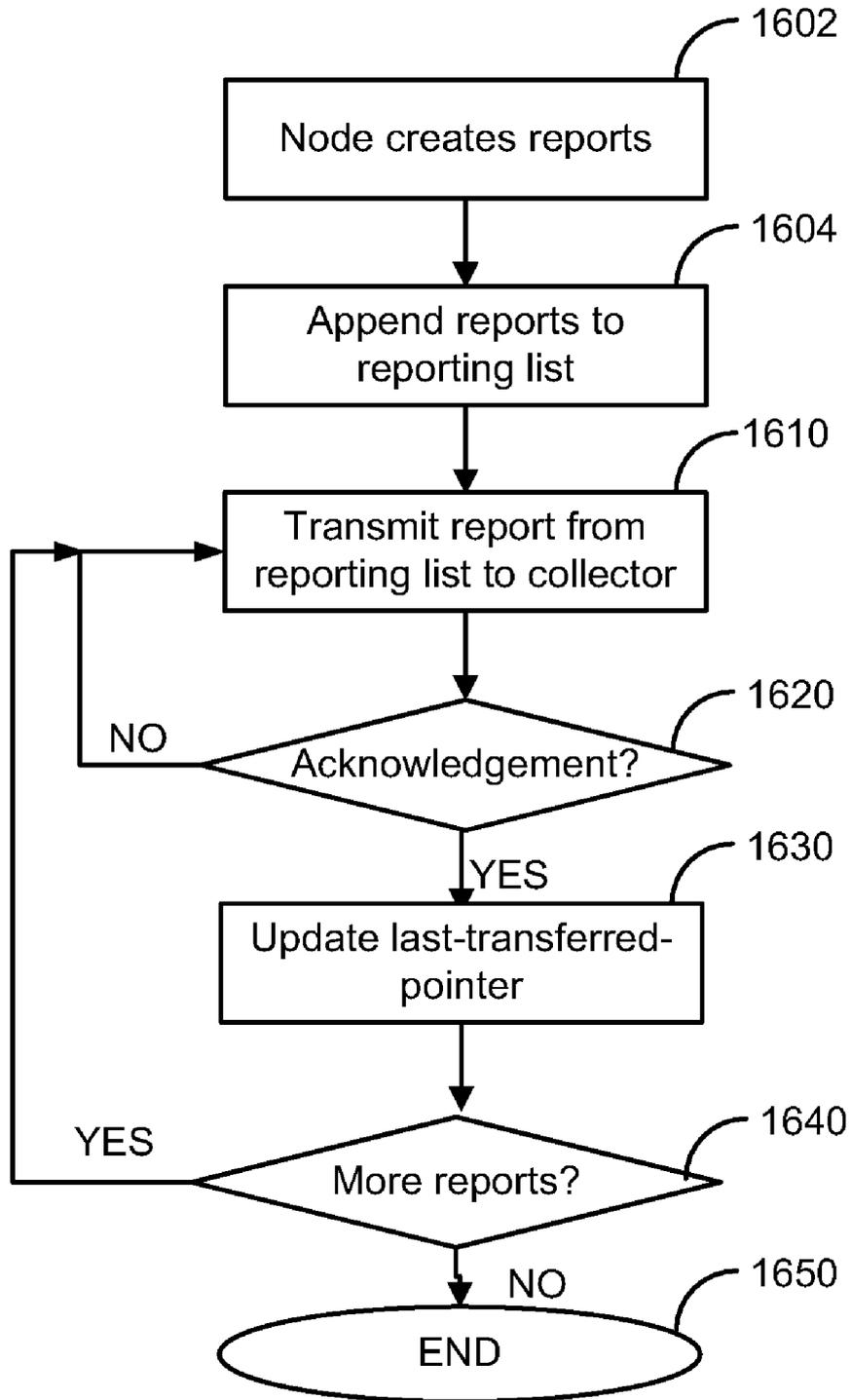


FIG. 16

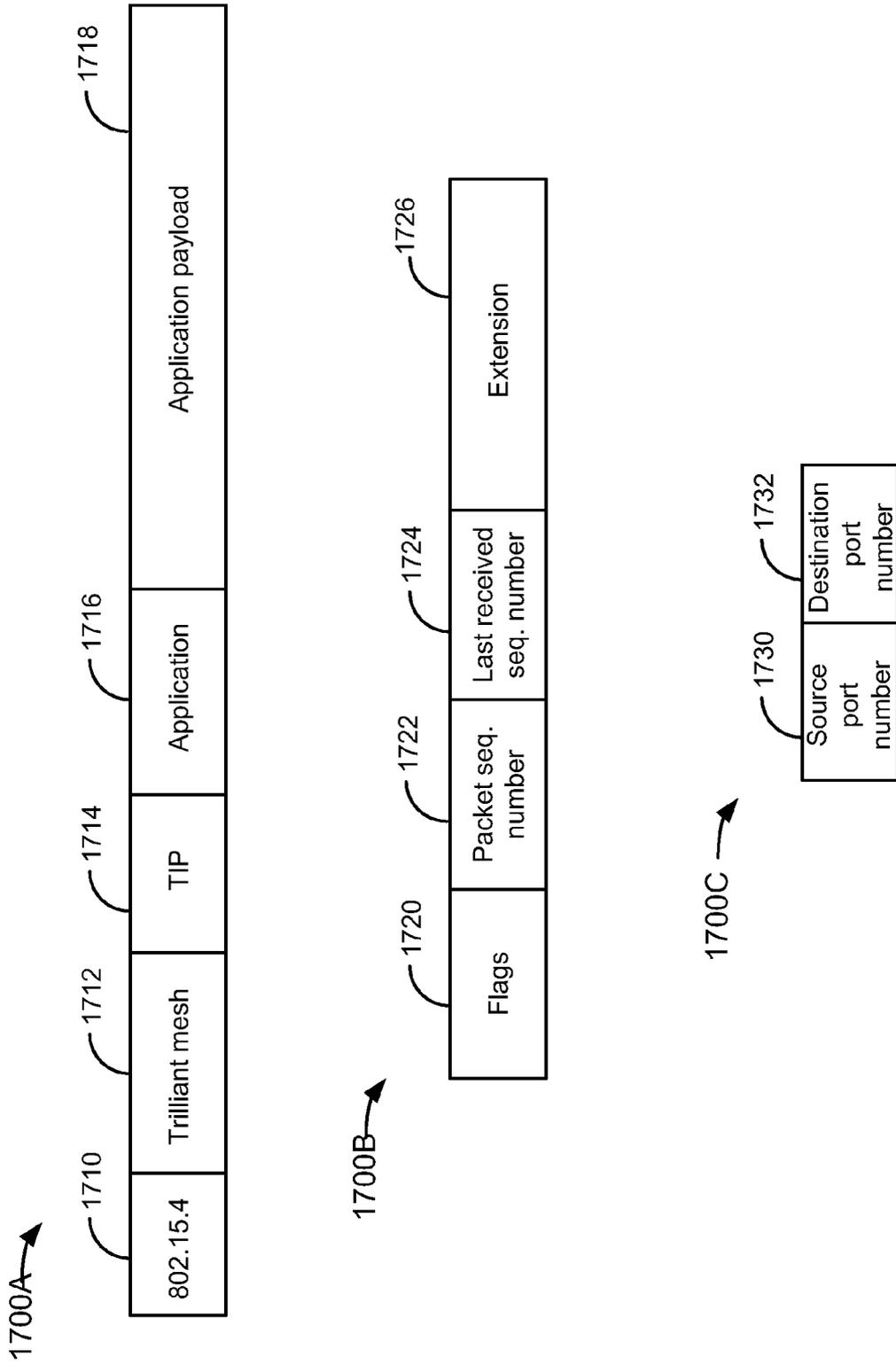


FIG. 17

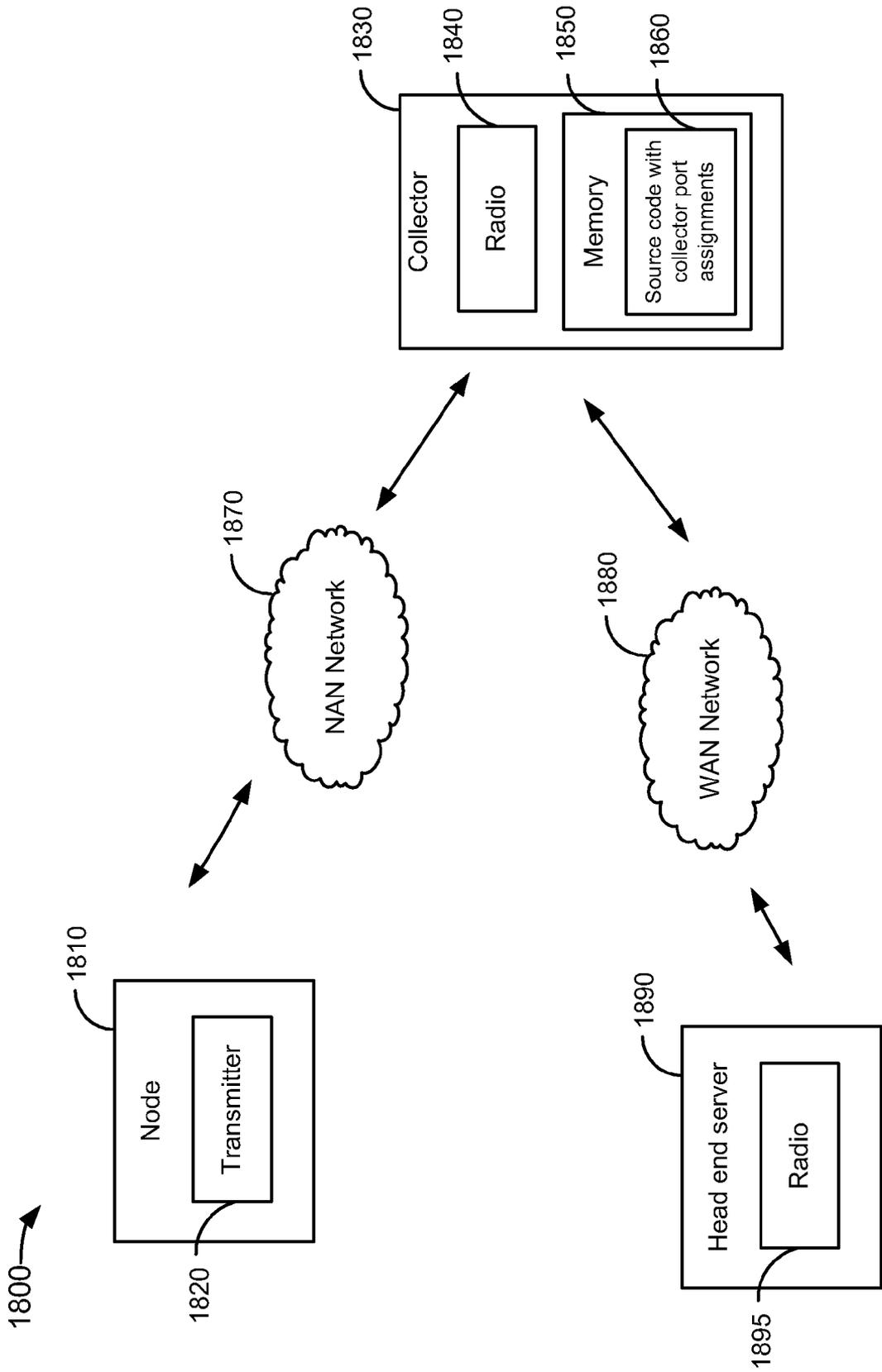


FIG. 18

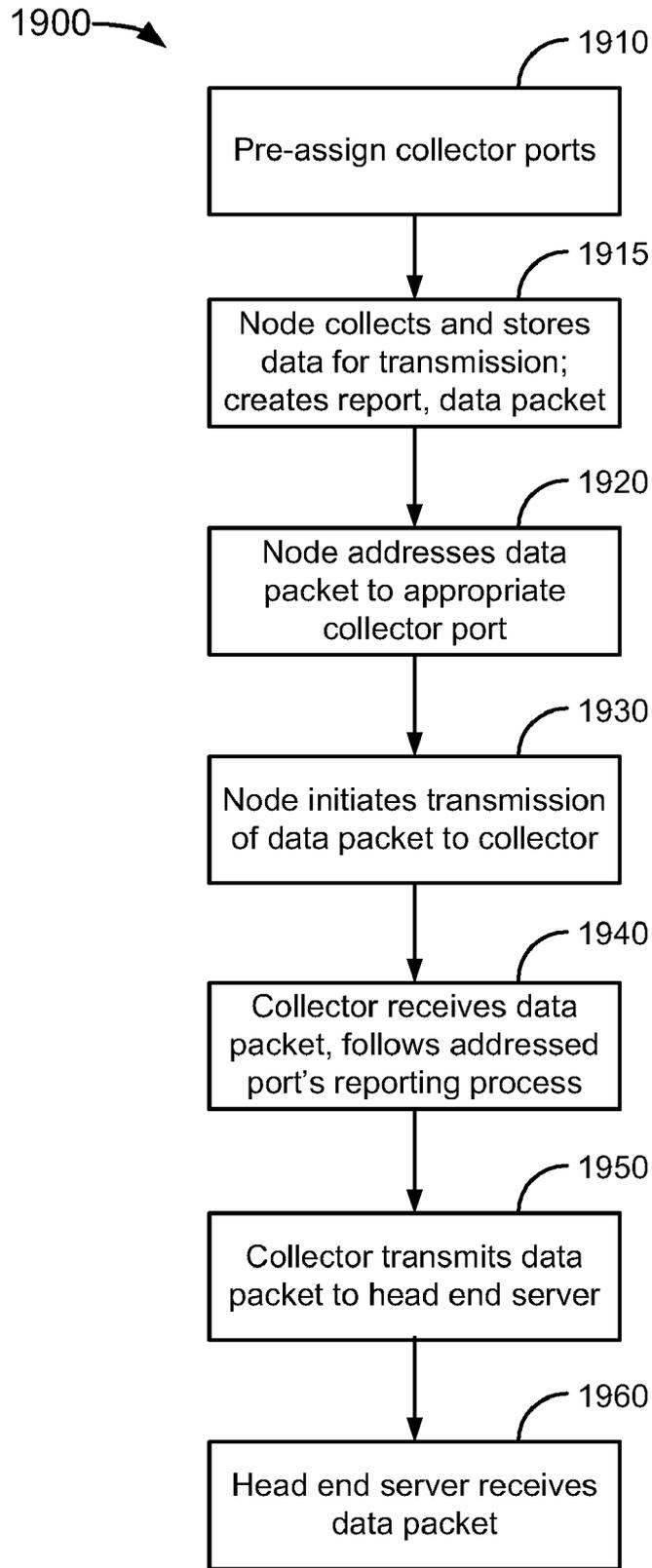


FIG. 19

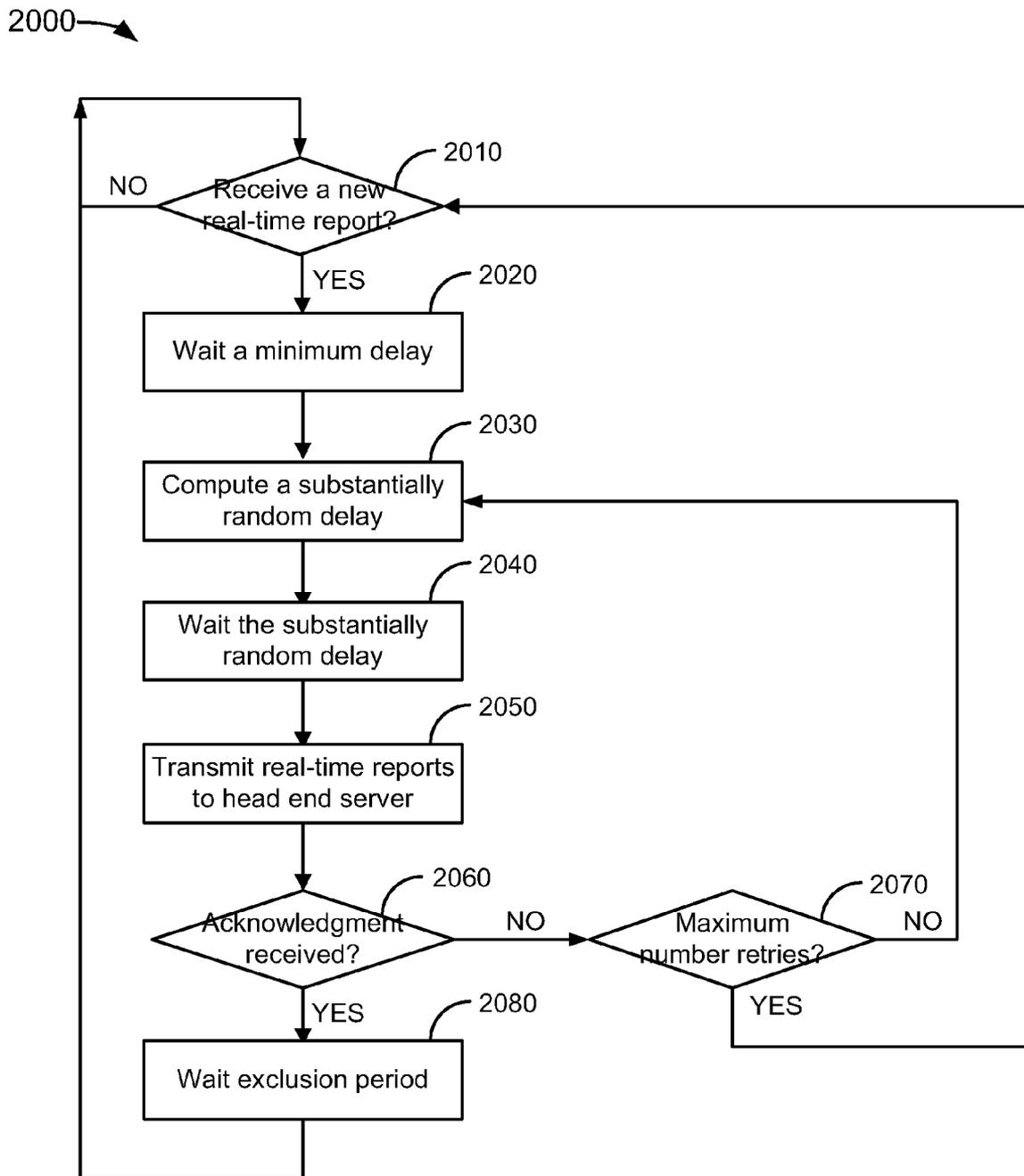


FIG. 20

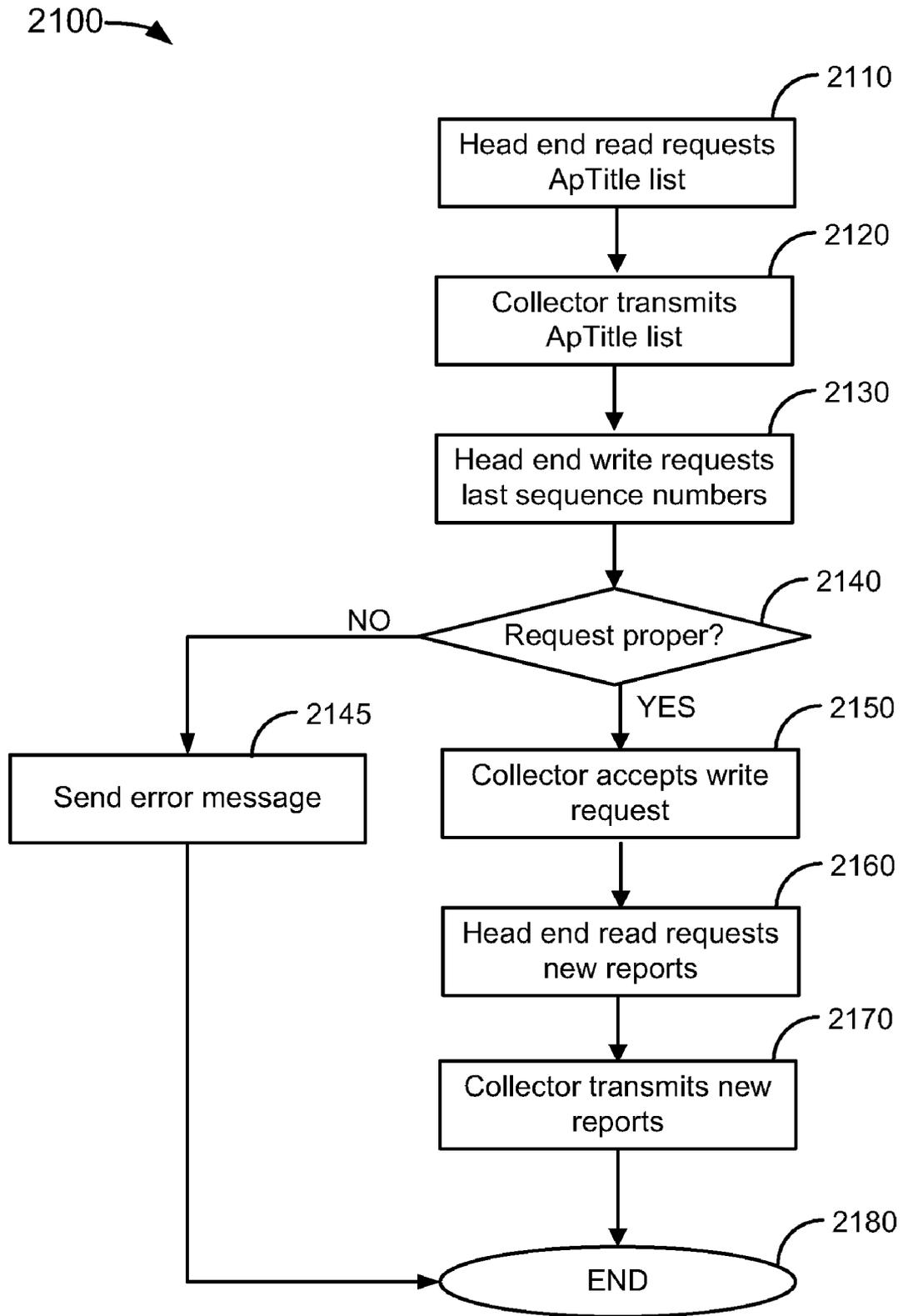


FIG. 21

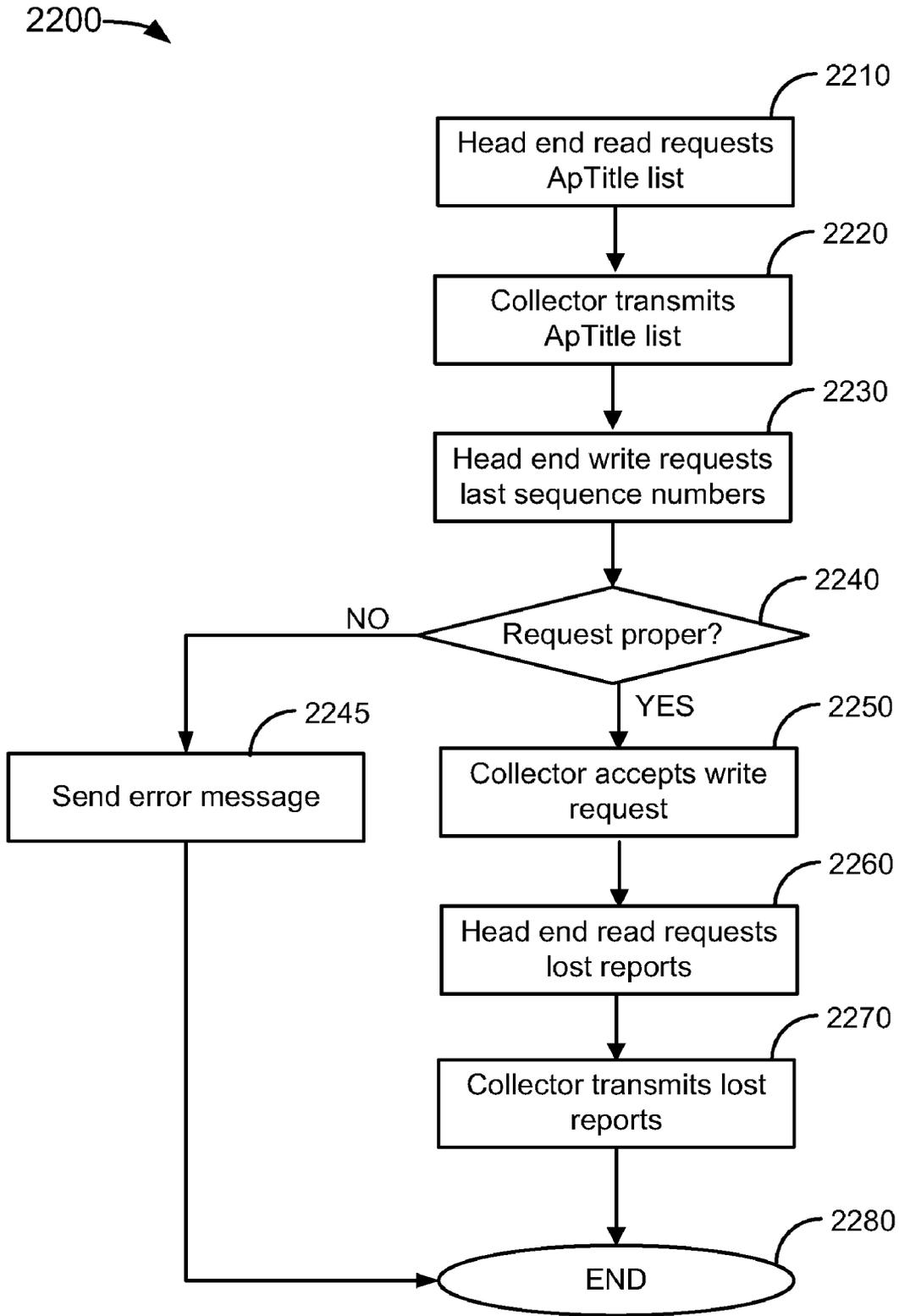


FIG. 22

**COLLECTOR DEVICE AND SYSTEM
UTILIZING STANDARDIZED UTILITY
METERING PROTOCOL**

**CROSS-REFERENCE TO RELATED
APPLICATIONS**

[0001] This application claims the benefit of priority to the following United States provisional patent applications which are incorporated herein by reference in their entirety:

- [0002] Ser. No. 60/989,957 entitled "Point-to-Point Communication within a Mesh Network", filed Nov. 25, 2007 (TR0004-PRO);
- [0003] Ser. No. 60/989,967 entitled "Efficient And Compact Transport Layer And Model For An Advanced Metering Infrastructure (AMI) Network," filed Nov. 25, 2007 (TR0003-PRO);
- [0004] Ser. No. 60/989,958 entitled "Creating And Managing A Mesh Network Including Network Association," filed Nov. 25, 2007 (TR0005-PRO);
- [0005] Ser. No. 60/989,964 entitled "Route Optimization Within A Mesh Network," filed Nov. 25, 2007 (TR0007-PRO);
- [0006] Ser. No. 60/989,950 entitled "Application Layer Device Agnostic Collector Utilizing ANSI C12.22," filed Nov. 25, 2007 (TR0009-PRO);
- [0007] Ser. No. 60/989,953 entitled "System And Method For Real Time Event Report Generation Between Nodes And Head End Server In A Meter Reading Network Including From Smart And Dumb Meters," filed Nov. 25, 2007 (TR0010-PRO);
- [0008] Ser. No. 60/989,975 entitled "System and Method for Network (Mesh) Layer And Application Layer Architecture And Processes," filed Nov. 25, 2007 (TR0014-PRO);
- [0009] Ser. No. 60/989,959 entitled "Tree Routing Within a Mesh Network," filed Nov. 25, 2007 (TR0017-PRO);
- [0010] Ser. No. 60/989,961 entitled "Source Routing Within a Mesh Network," filed Nov. 25, 2007 (TR0019-PRO);
- [0011] Ser. No. 60/989,962 entitled "Creating and Managing a Mesh Network," filed Nov. 25, 2007 (TR0020-PRO);
- [0012] Ser. No. 60/989,951 entitled "Network Node And Collector Architecture For Communicating Data And Method Of Communications," filed Nov. 25, 2007 (TR0021-PRO);
- [0013] Ser. No. 60/989,955 entitled "System And Method For Recovering From Head End Data Loss And Data Collector Failure In An Automated Meter Reading Infrastructure," filed Nov. 25, 2007 (TR0022-PRO);
- [0014] Ser. No. 60/989,952 entitled "System And Method For Assigning Checkpoints To A Plurality Of Network Nodes In Communication With A Device Agnostic Data Collector," filed Nov. 25, 2007 (TR0023-PRO);
- [0015] Ser. No. 60/989,954 entitled "System And Method For Synchronizing Data In An Automated Meter Reading Infrastructure," filed Nov. 25, 2007 (TR0024-PRO);
- [0016] Ser. No. 60/992,312 entitled "Mesh Network Broadcast," filed Dec. 4, 2007 (TR0027-PRO);
- [0017] Ser. No. 60/992,313 entitled "Multi Tree Mesh Networks", filed Dec. 4, 2007 (TR0028-PRO);

- [0018] Ser. No. 60/992,315 entitled "Mesh Routing Within a Mesh Network," filed Dec. 4, 2007 (TR0029-PRO);
 - [0019] Ser. No. 61/025,279 entitled "Point-to-Point Communication within a Mesh Network", filed Jan. 31, 2008 (TR0030-PRO), and which are incorporated by reference.
 - [0020] Ser. No. 61/025,270 entitled "Application Layer Device Agnostic Collector Utilizing Standardized Utility Metering Protocol Such As ANSI C12.22," filed Jan. 31, 2008 (TR0031-PRO);
 - [0021] Ser. No. 61/025,276 entitled "System And Method For Real-Time Event Report Generation Between Nodes And Head End Server In A Meter Reading Network Including Form Smart And Dumb Meters," filed Jan. 31, 2008 (TR0032-PRO);
 - [0022] Ser. No. 61/025,282 entitled "Method And System for Creating And Managing Association And Balancing Of A Mesh Device In A Mesh Network," filed Jan. 31, 2008 (TR0035-PRO);
 - [0023] Ser. No. 61/025,271 entitled "Method And System for Creating And Managing Association And Balancing Of A Mesh Device In A Mesh Network," filed Jan. 31, 2008 (TR0037-PRO);
 - [0024] Ser. No. 61/025,287 entitled "System And Method For Operating Mesh Devices In Multi-Tree Overlapping Mesh Networks", filed Jan. 31, 2008 (TR0038-PRO);
 - [0025] Ser. No. 61/025,278 entitled "System And Method For Recovering From Head End Data Loss And Data Collector Failure In An Automated Meter Reading Infrastructure," filed Jan. 31, 2008 (TR0039-PRO);
 - [0026] Ser. No. 61/025,273 entitled "System And Method For Assigning Checkpoints to A Plurality Of Network Nodes In Communication With A Device-Agnostic Data Collector," filed Jan. 31, 2008 (TR0040-PRO);
 - [0027] Ser. No. 61/025,277 entitled "System And Method For Synchronizing Data In An Automated Meter Reading Infrastructure," filed Jan. 31, 2008 (TR0041-PRO); and
 - [0028] Ser. No. 61/094,116 entitled "Message Formats and Processes for Communication Across a Mesh Network," filed Sep. 4, 2008 (TR0049-PRO).
- [0029] This application hereby references and incorporates by reference each of the following United States patent applications filed contemporaneously herewith:
- [0030] Ser. No. _____ entitled "Point-to-Point Communication within a Mesh Network", filed Nov. 21, 2008 (TR0004-US);
 - [0031] Ser. No. _____ entitled "Efficient And Compact Transport Layer And Model For An Advanced Metering Infrastructure (AMI) Network," filed Nov. 21, 2008 (TR0003-US);
 - [0032] Ser. No. _____ entitled "Communication and Message Route Optimization and Messaging in a Mesh Network" filed Nov. 21, 2008 (TR0007-US);
 - [0033] Ser. No. _____ entitled "Collector Device and System Utilizing Standardized Utility Metering Protocol," filed Nov. 21, 2008 (TR0009-US);
 - [0034] Ser. No. _____ entitled "Method and System for Creating and Managing Association and Balancing of a Mesh Device in a Mesh Network," filed Nov. 21, 2008 (TR0020-US); and

[0035] Ser. No. _____ entitled “System And Method For Operating Mesh Devices In Multi-Tree Overlapping Mesh Networks”, filed Nov. 21, 2008 (TR0038-US).

BACKGROUND OF THE INVENTION

[0036] In an advanced metering infrastructure (AMI), the metering infrastructure may include system and methodological components where the system measures, collects, and advantageously analyzes usage of utilities such as but not limited to electricity, gas, and water; communication resources and/or other utilities or services through the use of advanced metering devices; two-way communication networks; and data management systems.

[0037] American National Standards Institute (ANSI) C12 protocols have been established which standardize metering products and communications of metering data in the United States and North America generally, and other European and International protocols and standards have similarly been promulgated. For example, ANSI C12.18 establishes protocols for point-to-point meter communication connections; ANSI C12.19 establishes data structures as well as rules and syntax pertaining to the data structures; and ANSI C12.22 establishes the protocol standards for transporting data tables over a network. In particular, ANSI C12.22 PSEM (Protocol Specification for Electric Metering) establishes protocols for the pulling of data or reports from a meter at a node, where the node must receive a request for information before the node will transmit any information.

SUMMARY OF THE INVENTION

[0038] A device-agnostic collector may either request data or a report from a node, also referred to as pulling information from the node, or accept information pushed from a node when a node initiates the transmission of data or a report to the collector, rather than waiting for an information request. A node is a device with the ability to communicate across a network, such as for example across an AMI network, and devices include utility-measuring devices like meters and utility-consumption devices like appliances. The node transmits data structures according to the specified protocols, such as for example an ANSI C12 or other national, regional, or international standards.

[0039] When a device-agnostic collector receives pushed reports from a node, the device-agnostic collector does not need any special application programs for communicating with different types of devices about various types of data structures supported by the different devices; the device-agnostic collector only requires the ability to receive reports addressed and pushed to particular ones of its ports and to acknowledge the received reports. Individual ports of the device-agnostic collector may be reserved for receiving specific types of reports, such as batch-reporting and real-time reporting. Based upon the port to which a node addresses a report to the device-agnostic collector, the collector will process the report and transmit the report to the head end server appropriately.

[0040] In one aspect, embodiments provide a collector device and system utilizing standardized utility metering protocols.

[0041] In one aspect, embodiments provide a system for communicating data from a node to a head end server, the system comprising: a node for measuring, storing, and initiating transmission of data to a collector over a first network;

a collector for receiving the data over the first network and transmitting the data over a second network to the head end server; the head end server being operative for receiving the data over the second network.

[0042] In another aspect, embodiments provide a method of communicating data from a node to a head end server, the method comprising: collecting, storing, and initiating transmission of data by the node to a collector over a first network; receiving, by the collector, the data over the first network and transmitting, by the collector, the data over a second network to the head end server; and receiving, by the head end server, the data over the second network.

[0043] In another aspect, embodiments provide a collector device for operating in a system of the type that includes at least one node operative on a first network for storing data and a network server operative for receiving data on a second network, the collector device comprising: a receiver unit for receiving the data from the node over the first network; and a transmitter unit for transmitting data to the server over the second network; the collector being node device and data-structure agnostic and operative to: (i) request or pull data from a node device, or (ii) accept information pushed from a node device when a node device initiates the transmission of data or a report to the collector, or (iii) a combination of request or pull data from a node device and accept information pushed from a node.

[0044] In another aspect, embodiments provide a method of operating a collector in a network environment including a plurality of different networked node devices having different device characteristics and different data structure characteristics and a networked head end server, the method comprising: receiving a first data over the network environment from a first node device of a first node device type; receiving a second data over the network environment from a second node device of a second node device type; identifying the first data as the first data type and the second data as the second data type; processing the received first data and second data according to whether it is the first data type or the second data type to generate a processed data; and transmitting the processed data to the server over the network environment.

[0045] In another aspect, embodiments provide a computer program stored in a computer readable form for execution within a processor and processor associated memory for controlling the operation of a collector and of a system including the collector.

[0046] There are different types of reports that a node transmits to the head end server through the collector including batch reporting and real-time reporting. Real-time reports are needed urgently by the head end while batched reports are not. With both of these methods of reporting, a node may first transmit data packets to the collector. However, based upon the type of data, the node may address the data packets to specific ports of the collector. By pre-assigning different ports of the collector to receive different types of reports, the collector does not need to read and process the data packets. The collector will follow the appropriate reporting procedures for the particular type of report.

[0047] In one aspect, there is provided a method of transmitting a report from a node to a head end server, the method comprising: pre-assigning one or more ports at a collector to receive particular types of reports; transmitting a data packet from the node to the collector over a first network, wherein the data packet includes a header containing a destination port of the collector and the report; and transmitting the data packet

from the collector to the head end server over a second network, wherein the collector follows reporting procedures based upon the destination port.

[0048] In another aspect, there is provided a method of transmitting data from a smart meter to a communications card for transmission to the collector, the method comprising: storing one or more lists of data structures in a register at the smart meter; storing a sequence number for each list in the register at the smart meter, wherein the sequence number is incremented by a predetermined number when a new piece of data is appended to the list; accessing and downloading new data from the one or more lists to a communications card, wherein the new data has corresponding sequence numbers that have not been previously downloaded; storing the data and the corresponding sequence number in memory; creating one or more reports from the new data at the communications card; and transmitting the one or more reports to the collector, wherein the communications card initiates the transmission.

[0049] In another aspect, there is provided a method of using a reporting list at a node for transmission to a collector, the method comprising: creating one or more reports by the node; appending the one or more reports to a reporting list at the node; transmitting to the collector a report on the reporting list immediately following a flagged report flagged by a last-transferred-pointer, wherein the transmission is initiated by the collector; receiving an acknowledgement from the collector; and updating the last-transferred-pointer; wherein the last-transferred-pointer is used to determine reports that have not been received by the collector.

[0050] In another aspect, there is provided a system for transmitting a report from a node to a head end server, comprising: means for pre-assigning one or more ports at a collector to receive particular types of reports; means for transmitting a data packet from the node to the collector over a first network, wherein the data packet includes a header containing a destination port of the collector and the report; and means for transmitting the data packet from the collector to the head end server over a second network, wherein the collector follows reporting procedures based upon the destination port.

[0051] In another aspect, there is provided a system for transmitting a report from a node to a head end server, comprising: collector computer code stored in a collector memory for pre-assigning one or more ports at a collector to receive particular types of reports; a node transmitter for transmitting a data packet from the node to the collector over a first network, wherein the data packet includes a header containing a destination port of the collector and the report; and a collector transmitter for transmitting the data packet from the collector to the head end server over a second network, wherein the collector follows reporting procedures based upon the destination port.

[0052] In another aspect, there is provided a system for transmitting data from a smart meter to a communications card for transmission to the collector, comprising: means for storing one or more lists of data structures at the smart meter; means for storing a sequence number for each list at the smart meter, wherein the sequence number is incremented by a predetermined number when a new piece of data is appended to the list; means for accessing and downloading new data from the one or more lists to a communications card, wherein the new data has corresponding sequence numbers that have not been previously downloaded; means for storing the data and the corresponding sequence number; means for creating

one or more reports from the new data at the communications card; and means for transmitting the one or more reports to the collector, wherein the communications card initiates the transmission.

[0053] In another aspect, there is provided a system for transmitting data from a smart meter to a communications card for transmission to the collector, comprising: a register for storing one or more lists of data structures at the smart meter and a sequence number for each list at the smart meter, wherein the sequence number is incremented by a predetermined number when a new data structure is appended to the list; a processor for accessing data structures on the list and creating one or more reports from the new data at the communications card; a receiver and a memory for downloading the new data, wherein the new data has corresponding sequence numbers that have not been previously downloaded, and further wherein the memory stores the new data and the corresponding sequence number; and a node transmitter for transmitting the one or more reports to the collector, wherein the communications card initiates the transmission.

[0054] In another aspect, there is provided a system for using a reporting list at a node for transmission to a collector comprising: means for creating one or more reports by the node; means for appending the one or more reports to a reporting list at the node; means for transmitting to the collector a report on the reporting list immediately following a flagged report flagged by a last-transferred-pointer; means for receiving an acknowledgement from the collector; and means for updating the last-transferred-pointer; wherein the last-transferred-pointer is used to determine reports that have not been received by the collector.

[0055] In another aspect, there is provided a system for using a reporting list at a node for transmission to a collector comprising: a node processor for creating one or more reports by the node, appending the one or more reports to a reporting list at the node, and updating a last-transferred-pointer; a node transmitter for transmitting to the collector a report on the reporting list immediately following a flagged report flagged by the last-transferred-pointer; and a node receiver for receiving an acknowledgement from the collector; wherein the last-transferred-pointer is used to determine reports that have not been received by the collector.

[0056] In another aspect, there is provided a computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of transmitting a report from a node to a head end server, the method comprising: pre-assigning one or more ports at a collector to receive particular types of reports; transmitting a data packet from the node to the collector over a first network, wherein the data packet includes a header containing a destination port of the collector and the report; and transmitting the data packet from the collector to the head end server over a second network, wherein the collector follows reporting procedures based upon the destination port.

[0057] In another aspect, there is provided a computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of transmitting data from a smart meter to a communications card for transmission to the collector, the method comprising: storing one or more lists of data structures in a register at the smart meter; storing a sequence number for each list in the register at the smart meter, wherein the sequence number is incremented by a predetermined

number when a new piece of data is appended to the list; accessing and downloading new data from the one or more lists to a communications card, wherein the new data has corresponding sequence numbers that have not been previously downloaded; storing the data and the corresponding sequence number in memory; creating one or more reports from the new data at the communications card; and transmitting the one or more reports to the collector, wherein the communications card initiates the transmission.

[0058] In another aspect, there is provided a computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of using a reporting list at a node for transmission to a collector, the method comprising: creating one or more reports by the node; appending the one or more reports to a reporting list at the node; transmitting to the collector a report on the reporting list immediately following a flagged report flagged by a last-transferred-pointer, wherein the transmission is initiated by the collector; receiving an acknowledgement from the collector; and updating the last-transferred-pointer; wherein the last-transferred-pointer is used to determine reports that have not been received by the collector.

[0059] A device-agnostic collector can accept information pushed from a node where the node initiates the transmission of information to the collector. The collector does not require any special logic or programming about the node or the data structures transmitted by the node, thus the collector is device-agnostic. Nodes pushing information may be assigned random staggered checkpoints to prevent the nodes from transmitting their data packets simultaneously.

[0060] In one aspect, there is provided a node device for initiating transmission of data to a collector comprising: a data input/output module for collecting the data; a memory for storing the data; a processor for creating a report from the data; and a radio for transmitting the report to the collector, wherein the transmission is initiated by the node.

[0061] In another aspect, there is provided a collector for receiving a data packet over a first network initiated from a node comprising: a first radio for receiving the data packet initiated by the node over the first network; a second radio for transmitting the data packet over a second network to a head end server; and a memory for storing the data packet to be transmitted to the head end server and used by the processor.

[0062] In another aspect, there is provided a method of operating a node that initiates a report to a collector comprising: collecting data through an input device; storing the data in a memory; creating the report from the data; and transmitting the report to the collector over a first network, wherein the report is initiated by the node.

[0063] In another aspect, there is provided a method of operating a collector that receives a data packet initiated by a node comprising: receiving the data packet initiated by the node over a first network addressed to a particular collector port, wherein the data packet comprises a report; processing the report according to pre-assigned procedures corresponding to the particular collector port; and transmitting the report to a head end over a second network.

[0064] In another aspect, there is provided a computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of operating a node that initiates a report to a collector comprising: collecting data through an input device; storing the data in a memory; creating the report from

the data; and transmitting the report to the collector over a first network, wherein the report is initiated by the node.

[0065] In another aspect, there is provided a computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of operating a collector that receives a data packet initiated by a node comprising: receiving the data packet initiated by the node over a first network addressed to a particular collector port, wherein the data packet comprises a report; processing the report according to pre-assigned procedures corresponding to the particular collector port; and transmitting the report to a head end over a second network.

[0066] In one aspect, there is provided a method of communicating between a plurality of nodes and a collector using checkpoints, comprising: generating a substantially random time slot within a pre-scheduled reporting time window for each node; and transmitting data packets from each node to the collector at the node's substantially random time slot within the prescheduled reporting time window.

[0067] In another aspect, there is provided a method of transmitting data packets from a plurality of nodes to a collector, comprising: generating a first substantially random time slot within a prescheduled reporting time window for each node; transmitting data packets from each node to the collector at the node's substantially random time slot within the prescheduled reporting time window; transmitting an acknowledgement from the collector for each received data packet; generating a second substantially random time slot within a second prescheduled reporting time window for each node that does not receive an acknowledgement from the collector; and re-transmitting data packets from each node that did not receive the acknowledgement from the collector at the node's second substantially random time slot within the second prescheduled reporting time window.

[0068] In another aspect, there is provided a system for communicating between a plurality of nodes and a collector using checkpoints, comprising: means for generating a substantially random time slot within a pre-scheduled reporting time window for each node; and means for transmitting data packets from each node to the collector at the node's substantially random time slot within the prescheduled reporting time window.

[0069] In another aspect, there is provided a system for communicating between a plurality of nodes and a collector using checkpoints, comprising: a substantially random time slot generator for generating a substantially random time slot within a pre-scheduled reporting time window for each node; and a node transmitter for transmitting data packets from each node to the collector at the node's substantially random time slot within the prescheduled reporting time window.

[0070] In another aspect, there is provided a system for transmitting data packets from a plurality of nodes to one collector comprising: means for generating a first substantially random time slot within a prescheduled reporting time window for each node; means for transmitting data packets from each node to the collector at the node's substantially random time slot within the prescheduled reporting time window; means for generating a second substantially random time slot within a second prescheduled reporting time window at each node that does not receive an acknowledgement from the collector; and means for re-transmitting data packets from each node that did not receive the acknowledgement

from the collector at the node's second substantially random time slot within the second prescheduled reporting time window.

[0071] In another aspect, there is provided a system for transmitting data packets from a plurality of nodes to one collector comprising: a substantially random time slot generator for generating a first substantially random time slot within a prescheduled reporting time window for each node and for generating a second substantially random time slot within a second prescheduled reporting time window at each node that does not receive an acknowledgement from the collector; and a transmitter for transmitting data packets from each node to the collector at the node's substantially random time slot within the prescheduled reporting time window and for re-transmitting data packets from each node that did not receive the acknowledgement from the collector at the node's second substantially random time slot within the second prescheduled reporting time window.

[0072] In another aspect, there is provided a computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of communicating between a plurality of nodes and a collector using checkpoints, comprising: generating a substantially random time slot within a pre-scheduled reporting time window for each node; and transmitting data packets from each node to the collector at the node's substantially random time slot within the prescheduled reporting time window.

[0073] In another aspect, there is provided a computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of transmitting data packets from a plurality of nodes to a collector, comprising: generating a first substantially random time slot within a prescheduled reporting time window for each node; transmitting data packets from each node to the collector at the node's substantially random time slot within the prescheduled reporting time window; transmitting an acknowledgement from the collector for each received data packet; generating a second substantially random time slot within a second prescheduled reporting time window for each node that does not receive an acknowledgement from the collector; and re-transmitting data packets from each node that did not receive the acknowledgement from the collector at the node's second substantially random time slot within the second prescheduled reporting time window.

[0074] A three-step synchronization process is used to transfer aggregated reports from a collector to the head end server. The head end server first sends a read request for an ApTitle list, and the collector returns the list. Then the head end server sends a write request of last sequence numbers corresponding to each ApTitle, and the collector either accepts the request or returns an error due to an improper request. Finally, the head end server sends a read request for new reports, and the collector transmits the reports. The three-step process allows a collector to identify and send the reports which the head end server does not have stored at the head end database.

[0075] In one aspect there is provided a method of synchronizing a data request between a head end server and a collector comprising: transmitting from the head end server a first read request for a first table from the collector, wherein elements in the first table correspond to nodes serviced by the collector; transmitting the first table from the collector to the

head end server; transmitting from the head end server a write request, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last received report from each node stored in a database at the head end server; accepting the write request if the write request is proper; transmitting from the head end server a second read request for new reports; and transmitting the new reports from the collector to the head end server.

[0076] In another aspect there is provided a system for synchronizing a data request between a head end server and a collector comprising: means for transmitting from the head end server a first read request for a first table from the collector, wherein elements in the first table correspond to nodes serviced by the collector; means for transmitting the first table from the collector to the head end server; means for transmitting from the head end server a write request, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last received report from each node stored in a database at the head end server; means for accepting the write request if the write request is proper; means for transmitting from the head end server a second read request for new reports; and means for transmitting the new reports from the collector to the head end server.

[0077] In another aspect there is provided a system for synchronizing a data request between a head end server and a collector comprising: (a) a head end server transmitter for transmitting: a first read request for a first table from the collector, wherein elements in the first table correspond to nodes serviced by the collector; a write request, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last received report from each node stored in a database at the head end server; and a second read request for new reports; (b) a collector transmitter for transmitting: the first table to the head end server; and the new reports to the head end server; and (c) a collector receiver and a collector processor for accepting the write request if the write request is proper.

[0078] In another aspect there is provided a computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of synchronizing a data request between a head end server and a collector comprising: transmitting from the head end server a first read request for a first table from the collector, wherein elements in the first table correspond to nodes serviced by the collector; transmitting the first table from the collector to the head end server; transmitting from the head end server a write request, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last received report from each node stored in a database at the head end server; accepting the write request if the write request is proper; transmitting from the head end server a second read request for new reports; and transmitting the new reports from the collector to the head end server.

[0079] In another aspect there is provided a method of synchronizing a data request between a head end server and a collector comprising: transmitting from the head end server a first read request for a first table from the collector, wherein elements in the first table correspond to nodes serviced by the collector; receiving by the head end server the first table from the collector; transmitting from the head end server a write

request, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last received report from each node stored in a database at the head end server; transmitting from the head end server a second read request for new reports; and receiving the new reports from the collector.

[0080] In another aspect there is provided a method of synchronizing a data request between a head end server and a collector comprising: receiving by the collector from the head end server a first read request for a first table, wherein elements in the first table correspond to nodes serviced by the collector; transmitting the first table from the collector to the head end server; receiving by the collector from the head end server a write request, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last received report from each node stored in a database at the head end server; accepting by the collector the write request if the write request is proper; receiving by the collector from the head end server a second read request for new reports; and transmitting the new reports from the collector to the head end server.

[0081] In another aspect there is provided a computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of synchronizing a data request between a head end server and a collector comprising: transmitting from the head end server a first read request for a first table from the collector, wherein elements in the first table correspond to nodes serviced by the collector; receiving by the head end server the first table from the collector; transmitting from the head end server a write request, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last received report from each node stored in a database at the head end server; transmitting from the head end server a second read request for new reports; and receiving the new reports from the collector.

[0082] In another aspect there is provided a computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of synchronizing a data request between a head end server and a collector comprising: receiving by the collector from the head end server a first read request for a first table, wherein elements in the first table correspond to nodes serviced by the collector; transmitting the first table from the collector to the head end server; receiving by the collector from the head end server a write request, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last received report from each node stored in a database at the head end server; accepting by the collector the write request if the write request is proper; receiving by the collector from the head end server a second read request for new reports; and transmitting the new reports from the collector to the head end server.

[0083] A node stores reports on its reporting list for transmission to the collector at its next checkpoint. A last-transferred-pointer is used to point to the last report on the reporting list transmitted to and acknowledged by the collector. If the collector fails, the node may re-associate with a new collector and continue transmitting reports from the reporting list that immediately follow the one flagged by the last-trans-

ferred-pointer. This mechanism allows the node to continue to forward reports to the head end server, independent of the particular collector used.

[0084] Once reports are aggregated by the collector, a three-step synchronization process may be used to transfer the aggregated reports from the collector to the head end server. The three-step process allows a collector to identify reports that the head end server has not yet received. This process may be used to recover from partial database losses at the head end server from the collector as well as to accommodate multiple head end servers.

[0085] In one aspect, there is provided a method of recovering loss of data at a head end server's database comprising: transmitting a first read request from the head end server for a first table from the collector, wherein elements in the first table correspond to nodes serviced by the collector; transmitting the first table from the collector to the head end server; transmitting a write request from the head end server, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last available report from each node stored in a database at the head end server after the loss of data; accepting the write request if the write request is proper; transmitting from the head end server a second read request for lost reports; and transmitting the lost reports from the collector to the head end server.

[0086] In another aspect, there is provided a method of recovering from a collector failure comprising: seeking a new collector by a node associated with the failed collector; and transmitting one or more reports from the node's reporting list.

[0087] In another aspect, there is provided a system for recovering from a collector failure comprising: means for seeking a new collector by a node associated with the failed collector; and means for transmitting one or more reports from the node's reporting list.

[0088] In another aspect, there is provided a system for recovering loss of data at a head end server comprising: means for transmitting a first read request from the head end server for a first table from the collector, wherein elements in the first table correspond to nodes serviced by the collector; means for transmitting the first table from the collector to the head end server; means for transmitting a write request from the head end server, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last available report from each node stored in a database at the head end server after the loss of data; means for accepting the write request if the write request is proper; means for transmitting from the head end server a second read request for lost reports; and means for transmitting the lost reports from the collector to the head end server.

[0089] In another aspect, there is provided a system for recovering loss of data at a head end server comprising: a head end server transmitter for transmitting: a first read request for a first table from the collector, wherein elements in the first table correspond to nodes serviced by the collector; a write request, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last available report from each node stored in a database at the head end server after the loss of data; and a second read request for lost reports; and a collector transmitter for transmitting: the first table from the collector to the head end server; and the lost reports from the

collector to the head end server; and a collector receiver and a collector processor for accepting the write request if the write request is proper.

[0090] In another aspect, there is provided a computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of recovering loss of data at a head end server's database comprising: transmitting a first read request from the head end server for a first table from the collector, wherein elements in the first table correspond to nodes serviced by the collector; transmitting the first table from the collector to the head end server; transmitting a write request from the head end server, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last available report from each node stored in a database at the head end server after the loss of data; accepting the write request if the write request is proper; transmitting from the head end server a second read request for lost reports; and transmitting the lost reports from the collector to the head end server.

[0091] In another aspect, there is provided a computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of recovering from a collector failure comprising: seeking a new collector by a node associated with the failed collector; and transmitting one or more reports from the node's reporting list.

[0092] This Summary introduces concepts in a simplified form that are described more fully below in the Detailed Description. This Summary is not intended to identify key or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0093] FIG. 1 is an illustration showing a non-limiting exemplary embodiment of a network topology that may be used in conjunction with aspects of the invention.

[0094] FIG. 2 shows two examples of block diagrams of an exemplary node, according to an aspect of the invention.

[0095] FIG. 3 shows an example of an electrical block diagram of an exemplary mesh gate or collector, according to an aspect of the invention.

[0096] FIG. 4 shows an example of an electrical block diagram of an exemplary microportal, according to an aspect of the invention.

[0097] FIG. 5 shows an example of an electrical block diagram of a head end server, according to an aspect of the invention.

[0098] FIG. 6 shows two examples of block diagrams of exemplary nodes including a meter, according to an aspect of the invention.

[0099] FIG. 7 is a flow chart illustrating an example of a method of downloading new data not previously downloaded from a smart meter register to a communications card at the node, according to an aspect of the invention.

[0100] FIG. 8 is a flow chart illustrating an example of a method of installing a node, according to an aspect of the invention.

[0101] FIG. 9 is a flow chart illustrating an example of a method of associating with a new collector, according to an aspect of the invention.

[0102] FIG. 10 is a flow chart illustrating an example of a method of pulling a report from a dumb meter, according to an aspect of the invention.

[0103] FIG. 11 is a flow chart illustrating an example of a method of pulling a report from a smart meter, according to an aspect of the invention.

[0104] FIG. 12 is a flow chart illustrating an example of a method of pulling data from a smart meter, according to an aspect of the invention.

[0105] FIG. 13 is a flow chart illustrating an example of a method of pushing data to a collector from a node, according to an aspect of the invention.

[0106] FIG. 14 shows an example of a block diagram of a node communicating with a collector through a network, according to an aspect of the invention.

[0107] FIG. 15 is a flow chart illustrating an example of a method of temporally staggering communications from the nodes to the collector, according to an aspect of the invention.

[0108] FIG. 16 is a flow chart illustrating an example of a method of tracking reports received by a collector, according to an aspect of the invention.

[0109] FIG. 17 shows examples of a packet data format, according to an aspect of the invention.

[0110] FIG. 18 shows an example of a block diagram of a system for transmitting a report from a node to a head end server, according to an aspect of the invention.

[0111] FIG. 19 is a flow chart illustrating an example of a method of informing a collector about which category a report falls into, according to an aspect of the invention.

[0112] FIG. 20 is a flow chart illustrating an example of a method of temporally staggering when a collector transmits real-time reports to the head end server, according to an aspect of the invention.

[0113] FIG. 21 is a flow chart illustrating an example of a method of transmitting reports for batch reporting between a collector and a head end server, according to an aspect of the invention.

[0114] FIG. 22 is a flow chart illustrating an example of a method of recovering lost reports at a head end server, according to an aspect of the invention.

DETAILED DESCRIPTION OF THE INVENTION

[0115] Described in detail below is a collector suitable for use in an advanced metering infrastructure and in other applications. The collector may be device-agnostic and data structure-agnostic because a device-agnostic and data structure-agnostic collector may receive reports pushed from devices at nodes without needing any special logic to understand the particular type of meter or device residing at the node or the data structure pushed by the node. Its ability to receive reports, such as but not limited to, data formatting information, utility meter readings, and consumption statistics, without needing special logic about the transmitting node is derived at least in part by the use of standardized report formats and port addressing by the node to direct particular types of reports to a specific port of the collector as described in greater detail hereinafter.

[0116] Various aspects of the embodiments will now be described. The following description provides specific details for a thorough understanding and enabling description of these examples. Many of these specific details are optional. One skilled in the art will understand, however, that the invention and its various embodiments may be practiced without many of these specific details and options. Additionally, some

well-known structures or functions may not be shown or described in detail, so as to avoid unnecessarily obscuring the relevant description.

[0117] The terminology used in the description presented below is intended to be interpreted in its broadest reasonable manner, even though it is being used in conjunction with a detailed description of certain specific examples. Certain terms may even be emphasized below; however, any terminology intended to be interpreted in any restricted manner will be overtly and specifically defined as such in this Detailed Description section.

[0118] Section headers and/or sub-headers are provided merely to guide the reader and are not intended to limit the scope of the invention in any way. Aspects, features, and elements of the invention and of embodiments of the invention are described throughout the written description and the drawings and claims.

[0119] Exemplary Network Topology In accordance with at least one non-limiting embodiment, an advanced metering infrastructure (AMI) network may span large geographical areas. In order to provide communications between nodes at a utility user's site and the head end server of the utility which processes usage data, the AMI may advantageously include, but is not limited to, the following elements: a head end server **160**, also referred to as a system head end server, server, head end, or back end; a wide area network **110**; a mesh gate **140**, also referred to as a NAN-WAN gate or an access point; a neighborhood area network or NAN **120**; and a microportal **150** which may more generically be referred to as a micro access portal for a residential premise or a home area network (HAN). A microportal device that is supported by the ZigBee Alliance is referred to as an energy service portal (ESP), however the ESP utilizes a different protocol. An AMI network may include one or more of any of these network elements, some of which are optional.

[0120] In the example of FIG. 1, one exemplary non-limiting embodiment of an AMI network topology **100** is shown. This example of an AMI network includes three networks (or sub-networks), a wide area network (WAN) **110**, a neighborhood area network (NAN) **120** sometimes also referred to as a local area network (LAN), and a home area network (HAN) **130** sometimes also referred to as a premise area network (PAN). The WAN **110** typically uses TCP/IP communications protocol and provides a way for the head end server **160** to communicate with the mesh gate **140** or NAN-WAN gate. It may be appreciated that the head end server may be one or multiple computers and more usually multiple computers operating together to provide capacity and redundancy. The head end server hardware and connectivity may be any hardware and connectivity known in the art or convenient. The NAN **120** may advantageously and typically adhere to IEEE 802.15.4 protocol and provides a way for a mesh gate **140** to communicate with NAN-associated nodes **121**, **122**, **123** in its sub-network or microportals **150** that are serviced by the NAN **120**; however, other different protocols may be utilized. The HAN **130** uses a communications protocol, such as for example a protocol established by the ZigBee Alliance and provides communications between a microportal **150** and HAN-associated nodes **131**, **132**, **133** that are serviced by the HAN **130**. The ZigBee Alliance is an association of companies working to enable reliable, cost-effective, low-power, wirelessly networked, monitoring and control products based on an open global standard. The ZigBee Alliance focuses on defining the network, security, and application software lay-

ers, providing interoperability and conformance testing specifications, promoting the ZigBee brand globally to build market awareness, and managing the evolution of the technology. It will be apparent to a person of skill in the art that more nodes (or fewer nodes) than those shown in FIG. 1 may be serviced by the NAN and the HAN. In fact, hundreds or thousands of nodes may communicate over a single NAN, and this will typically be the case for actual AMI infrastructures and systems.

[0121] Various communications protocols may be supported and used by the elements of the AMI network **100** to communicate over the WAN **110**, the NAN **120**, and the HAN **130**. However, it will be appreciated that for purposes of convenience, specific examples of communications protocols and standards may be indicated, particularly for aspects of the invention relative to a device-agnostic-collector.

[0122] NAN-associated nodes serviced by a NAN **121**, **122**, **123** communicate with a mesh gate **140** or NAN-WAN gate through the NAN **120**. Nodes may, for example, include a utility meter for a residential or commercial site or a utility-measuring device or a utility consumption device which includes, but is not limited to, thermostats and appliances within the residential or commercial site. FIG. 2 shows two examples of block diagrams of an exemplary node. FIG. 2A illustrates a functional block diagram **200A** of an exemplary generic node, and FIG. 2B illustrates an electrical block diagram **200B** of an exemplary node.

[0123] With reference to FIG. 2A, the block diagram **200A** shows three functional blocks of an exemplary generic node (**121**, **122**, **123**, **131**, **132**, **133** as shown in FIG. 1): an input/output device **210**, a table **212**, and a communications unit **214**. For example, if the node, e.g., **121**, **122**, **123**, **131**, **132**, **133** of FIG. 1, has a utility consumption and/or monitoring device such as a thermostat, then the input/output device **210** might comprise a keyboard or keypad for operating or enabling different display screens, for selecting different items to display, for controlling the thermostat, and for otherwise providing an interaction. Other input/output devices such as voice command input and speakers or other human interface devices may alternatively or additionally be utilized. Table **212** would correspond to data structures compiled by the thermostat based upon temperature readings, and the communications unit **214** would be the means with which the node communicates with the collector (such as for example with a mesh gate) using ANSI C12 protocol or other protocols that may be implemented, such as a communications card.

[0124] Alternatively, if the node has a utility sensing device such as a utility meter, the input/output device **210** may comprise a sensor for detecting utility usage. Table **212** may comprise registers in which the sensor data is stored, and communications unit **214** may comprise a communications card by which the meter at the node communicates with a collector.

[0125] With reference to FIG. 2B, the electrical block diagram **200B** is representative of an exemplary node which includes a power regulator **235**, an energy storage device **240**, a microcontroller **245**, a data input/output device or module **250**, a flash memory **255**, a real-time clock **260**, and a radio **265**.

[0126] The power regulator **235** may regulate the primary power supply for the node as well as the energy provided by the energy storage device **240** which acts as a temporary back-up or secondary power supply should the main power supply experience an outage.

[0127] The energy storage device **240** is a secondary power supply that provides the node with least a few extra minutes of power so that it can transmit previously un-transmitted reports to the collector. In some embodiments, the node may not include an energy storage device **240**.

[0128] The energy storage device **240** may be a battery, a capacitor, or a small, compact but high capacity super-capacitor. Battery power supplies may be used, but these are not advantageous as they have limited lifetimes and would require field servicing. Capacitors on the other hand may provide practically unlimited lifetime and provide a power source that is sufficient to maintain operation and transmit information to the collector when charged even if the normal power is interrupted. In one non-limiting embodiment of the invention, a super capacitor has a storage capacity of 50 farads with a starting output voltage of 2.7 volts and a physical package that is cylindrical with a diameter of 18 mm and a length of 40 mm. A super capacitor of this type is made by NESSCAP Co., Ltd. in Soowon, Korea and may be used.

[0129] The microcontroller **245** controls the functions of the node including, but not limited to, creating reports from data taken by the sensor, transmitting reports to the collector, and responding to queries by the head end server or the collector. The microcontroller **245** may be implemented as one unit or as a plurality of or several units, with each unit controlling one (or even more than one) of the functional aspects of the node when plural units are provided, the input/output device **210**, the table **212**, and the communications unit **214**. Other non-limiting embodiments may distribute functional aspects of the node to a plurality of units, and the division of functions may be in any combination.

[0130] The data input/output device **250** in the electrical block diagram **200B** of FIG. 2B has similar functionality as the input/output device **210** as described for the functional block diagram **200A** of FIG. 2A.

[0131] The non-volatile flash memory **255** stores reports and other data which the node does not want to lose if both the main power supply and any secondary power supply should fail including when the capacitor or super-capacitor may not store sufficient energy to carry out intended operations.

[0132] If the one or more power supplies to the node should fail, the real-time clock **260** receiving power from an internal capacitor during this time acts as a counter to determine how much time elapsed between the time the power supplies to the node ceased providing power until the restoration of power. Consequently, upon restoration of power, the node can return to normal operations such as taking data at scheduled intervals and transmitting reports to the collector as scheduled. In one embodiment, the real-time clock may continue to count and provide information regarding elapsed time for between for up to 14 days when a 1 microfarad capacitor is employed.

[0133] The radio **265** provides a means by which the NAN-associated node **121**, **122**, **123** communicates with the collector **140** and other NAN-associated nodes **121**, **122**, **123** through the NAN using a communications protocol, such as the afore described IEEE 802.15.4 and ANSI C12 protocols or other regional, international, or ISO protocols. Alternatively, the radio **265** provides a means by which the HAN-associated node **131**, **132**, **133** communicates with the microportal **150** and other HAN-associated nodes **131**, **132**, **133** through the HAN using a communications protocol, such as the afore described ZigBee Alliance protocols or other regional, international, or ISO protocols. The radio **265**

includes both a transmitter for transmitting data packets and a receiver for receiving data packets.

[0134] FIG. 3 is an illustration showing an example of an electrical block diagram **300** for an access point such as a NAN-WAN gate and also described as a mesh gate **140** in this and in related applications. The terms may be used interchangeably. The mesh gate or NAN-WAN gate may perform any one or more of many different functions including for example, but not limited to, one or any combination of: relaying information from a server (such as from a head end server) to the mesh network nodes, routing information, collecting information from the nodes and microportals within any sub-network that may be configured for transmission to a server (such as to the head end server), acting as a HAN coordinator, acting as a NAN-WAN gate, transmitting firmware upgrades, and/or multicasting messages. A mesh gate **140** may be referred to herein as a collector when focusing upon the mesh gate's collector function because among the mesh gate's many functions, it collects information from the NAN-associated nodes **121**, **122**, **123** and/or microportals **150** in its sub-network.

[0135] While a second access point's sub-network, such as a second mesh gate's sub-network, may overlap the geographical area of a first mesh gate's sub-network, each node and microportal is advantageously only associated with a single mesh gate at any given time, where association is a relationship that enables communication of information between two associated entities and may optionally comport with the ANSI protocols for association. In at least one non-limiting embodiment, this or these associations may change over time. Each mesh gate's network advantageously operates at a different radio frequency (RF) from other mesh gate's networks which may be either overlapping in extent or nearby. The NAN-associated nodes **121**, **122**, **123** communicate with the mesh gate **140** through the Neighborhood-Area Network (NAN) **120**, and the head end server **160** communicates with the mesh gate **140** through the Wide-Area Network (WAN) **110**.

[0136] Although the example AMI network **100** only shows one mesh gate **140**, in an alternative, any number of mesh gates **140** may be deployed. The mesh gate **140** may provide a gateway between the NAN network **120** and the head end server **160**.

[0137] The mesh gate **140** includes a central processing unit (CPU) **345** or other processor or processing logic and coupled memory, a WAN radio **350**, a NAN radio **355**, at least one and advantageously multiple communication connections such as Ethernet connections **360**, Random Access Memory (RAM) **330**, flash or other non-volatile memory **335**, a real-time clock **340**, an optional power regulator **320** where power regulation may be advantageous, and a battery **325** or other energy storage means. The mesh gate **140** communicates with the NAN-associated nodes **121**, **122**, **123** and microportal **150** through the NAN radio **355** using a communications protocol such as IEEE 802.15.4 and ANSI C12 communications protocol or other protocols as may be implemented.

[0138] The NAN radio **355** includes both a transmitter or other means for transmitting data packets and a receiver or other means for receiving data packets.

[0139] The WAN radio **350**, which advantageously operates under TCP/IP protocol but which may use other standard or non-standard or proprietary protocol, is used to communicate with the head end server **160** through the WAN **110**. The

WAN radio 350 includes both a transmitter or other means for transmitting data packets and a receiver or other means for receiving data packets. The mesh gate's CPU or other processor or processing logic 345 runs applications, such as one or more application functions for making information queries of data from the nodes, storing the data received from the nodes in memory, processing data aggregated from various nodes, and sending commands and messages from the head end server to the nodes, among other functions. The CPU or other processor or processing logic 345 may communicate with other devices or systems, such as with computers and/or servers through the Ethernet or other connections 360.

[0140] The flash or other non-volatile memory 335 stores reports and/or other data which the mesh gate 140 does not want to lose if both the main power supply and any secondary power supply such as the battery 325 or other storage means should fail or become exhausted. The Random Access Memory (RAM) 330 may be a volatile memory storage used by and coupled with the CPU or other processor or processing logic 345 as is known in the art.

[0141] The power regulator 320 regulates the primary power supply for the mesh gate 140. If the one or more power supplies to the mesh gate should fail, the real-time clock 340 acts as a counter to determine how much time elapsed between the time the power supplies to the mesh gate ceased providing power until the restoration of power, thus allowing the mesh gate to return to normal operations upon restoration of power.

[0142] A mesh gate or collector 140 may be placed at many different locations including, but not limited to, on a street light, on a telephone pole, at a socket behind a utility meter, and/or at virtually any location at which it has access to electrical power. The collector 140 aggregates and stores information from the NAN-associated nodes 121, 122, 123 and microportal 150 for transmission to the head end server 160 and also provides communications between the NAN-associated nodes 121, 122, 123 and microportal 150 and the head end server 160. Essentially, the last several hundred meters of the AMI infrastructure farthest from the head end server 160 and closest to the utility's customers are advantageously served and controlled by a collector 140.

[0143] HAN-associated nodes 131, 132, 133, advantageously serviced by the HAN 130, communicate with a microportal 150 through the HAN 130. A microportal 150 is located between a HAN 130 and a NAN 120 but need not share a node with a meter. Microportals 150 typically communicate with the HAN-associated nodes 131, 132, 133 within the premises (residence or commercial premises) of one customer, and multiple microportals may be present within one collector's sub-network. The restriction to a single premises may be for reasons of security, regulatory requirements, or for other reasons. Signals received by a microportal 150 through the NAN 120 from the collector 140 are distributed to the appropriate HAN-associated node through the HAN 130, and signals received by a microportal 150 through the HAN 130 from the HAN-associated nodes 131, 132, 133 are distributed to the collector 140 through the NAN 120 for transmission to the head end server 160. For example, if a utility wants to send a text or other message to a home display of one of the utility's users, the head end server 160 sends the text or other message to the collector 140 on the appropriate sub-network to which the user's home display device is coupled. The collector 140 transmits the text message through the NAN 120 to the appropriate microportal 150

which communicates with the intended recipient of the message. The microportal 150 will know if one or more displays are present in the home and distribute the message to those displays. The locations of the displays in the user's home include, but are not limited to, a thermostat or an in-home display device. A thermostat may conveniently be utilized as at least one such thermostat may typically be present in every residence or premises.

[0144] FIG. 4 shows an example of an electrical block diagram 400 of an exemplary microportal 150. A microportal is a control device that receives control commands from the head end server 160 and/or mesh gate 140 using a first protocol and translates the control commands and provides some contextual distribution to the HAN-associated nodes 131, 132, 133 in a second protocol. The microportal 150 may include a microcontroller 420, a first radio such as a ZigBee radio 430, a second radio such as a non-ZigBee radio such as a mesh NAN radio (or a WAN radio or a PAN radio) 440, a flash memory or other non-volatile memory 450, and a real-time clock 460 or other device for counting or measuring an elapsed time. The microportal 150 communicates with the collector 140 through the second or NAN radio (or a WAN radio) 440 using IEEE 802.15.4 and ANSI C12 communications protocol or other standard communications protocols as may be known in the art. The NAN radio 440 includes both a transmitter for transmitting data packets and a receiver for receiving data packets. The first or ZigBee radio 430, which operates under the first communications standards, such as for example the ZigBee Alliance communication standards, is used to communicate with HAN-associated nodes 131, 132, 133 coupled to the HAN 130. The ZigBee radio 430 includes both a transmitter for transmitting data packets and a receiver for receiving data packets.

[0145] It may be appreciated that in one non-limiting embodiment the two radio configurations may include one Zigbee type radio operating at its radio frequency and according to its protocol, as well as a second non-Zigbee radio that operates over the selected HAN, NAN, or WAN frequency and protocol.

[0146] The microcontroller 420 runs or executes applications for the microportal 150 which may include, but are not limited to, transmitting messages from the HAN-associated nodes 131, 132, 133 on the HAN 130 to the collector 140, and transmitting messages from the collector 140 to HAN-associated nodes 131, 132, 133 on the HAN 130. The flash or other non-volatile memory 450 stores reports and other data which the microportal does not want to lose if its power supply should fail or become exhausted. If the power supply should fail, the real-time clock 460 acts as a counter to determine how much time elapsed between the time the power supply to the microportal 150 ceases providing energy or power until the restoration of energy or power. Thus, the microportal 150 returns to normal operations upon restoration of power.

[0147] In one embodiment, the microportal may share the same location and primary power source as a node coupled to the HAN 131, 132, 133, such as a meter or thermostat for the residential or commercial site. In this case, a single radio (or so called single-sided radio) of the node may be replaced by a dual-radio (also referred to as a two-sided radio) in order to convert the node to a microportal without having to write new firmware or implement new hardware. The functionality of the microportal at the node is or may be hidden from the hosting device and the hosting device need not have this knowledge. A normal single sided radio may be replaced by a

portal radio (also referred to as a micro access portal). In one non-limiting embodiment, the two radios of the dual radio may be on the same side or on different sides of a common communication card or circuit PC board. In one non-limiting embodiment, a portal is a radio plus or in combination with a portal application processing logic, processor, or CPU. More generally, a portal, micro-portal, micro access portal and the like is a node on the mesh network.

[0148] Alternatively, the microportal may be a stand-alone device in a location separate from any other node. In this embodiment, the microportal would not reside topologically at any of the nodes coupled to the HAN 131, 132, 123. That is, the microportal 150 would have a different address from all other nodes serviced by the HAN. Thus, if any of the HAN-associated nodes coupled to the HAN 131, 132, 133 experiences a power outage, the other nodes may still communicate with the collector 140 and head end server 160 through the microportal 150.

[0149] The head end server 160 of the system is responsible for processing usage data received from the NAN-associated and HAN-associated nodes 121, 122, 123, 131, 132, 133 through the collector 140. FIG. 5 shows an example of a functional block diagram 500 for a head end server 160. Although the example AMI network 100 only shows one head end server 160, in an alternative, any number of head end servers 160 may be deployed. For example, the head end servers 160 may be distributed by geographical location for shorter communication distances and latency times. Alternatively and in addition, each utility accessing the AMI network may have one or more head end servers 160.

[0150] The head end server 160 may be one or a plurality of computing device(s) configured to receive information, such as meter readings, from a plurality of networks 100 and NAN-associated and HAN-associated nodes 121, 122, 123, 131, 132, 133. Typically the head end server 160 may be one or multiple computers and more usually multiple computers operating together to provide capacity and redundancy. The head end server 160 may also be configured to transmit instructions to the networks 100, mesh gates 140, and nodes 121, 122, 123, 131, 132, 133. The head end server 160 includes application programs for a master relay 520, a key manager 530, a task scheduler 550, a load balancer 560, running a stack 180, and other applications 540 which perform typical server functions not required to enable the current invention. The head end server 160 also includes a network interface 570. In one non-limiting embodiment, the network interface 570 may be an Internet interface.

[0151] In one non-limiting embodiment, the processing logic such as a MCU or CPU receives calendar information or data from or through the WAN, HAN, or PAN side radio directly or indirectly from the head end server 160. This calendar information may for example include dynamic pricing information, demand usage information or data, or other information that may pertain to days of the week, hours of the day, and/or other information or data that may pertain to utility usage. For example, the information may specify electricity pricing on an hour by hour basis for the next 24 hour time period. This pricing may be programmed into a time of use calendar. The non-Zigbee side (e.g., WAN, NAN, PAN side) implements and understands calendar functionality. The Zigbee side does not provide such calendar functionality, and the Zigbee standard is set to a point that it does not have any practical ability to be modified to provide such functionality. Furthermore, the network may provide an ability to identify

and use a multicast address on the mesh network, such as for example in a load control application, but Zigbee does not have or understand the notion of multi-cast address.

[0152] On the other hand, the Zigbee radio is adapted to communicate with the Zigbee radio in the Zigbee devices. The Zigbee side may send messages to Zigbee devices based on the information from the WAN side. Although the Zigbee radio and the non-Zigbee network radio (e.g., WAN, HAN, PAN, or the like) are separate radios, they both interface to or through a common communications circuit or card so that they may share information. In one embodiment, the two radios on a single communications card or circuit share a same processing logic unit (such as a MCU or CPU) and in another non-limiting embodiment, they have separate processing logic units that communicate with each other through an interface.

[0153] The master relay 520 stores and keeps track of addresses of nodes registered with the head end server 160. The process for registration of a node according to ANSI C12.22 protocol is described below.

[0154] The key manager 530 stores and looks up or queries encryption keys used to provide secure communications across the components of the AMI. Different types of keys may provide different sets of access rights. For example, a node 121, 122, 123, 131, 132, 133 may encrypt a report which only the head end server 160 has the key to decrypt the report.

[0155] The task scheduler 550 schedules the launch of applications or scripts at pre-defined times or after specific time intervals or according to other predetermined or dynamically determined events or conditions.

[0156] The load balancer 560 optimizes resource utilization at the head end server 160 and although optional is advantageously provided.

[0157] The head end server 160 may also execute many other applications 540 which are not listed here explicitly.

[0158] The head end server 160 also advantageously has multiple network interface connections 570, such as Internet Protocol (IP) based connections 570. These connections permit the head end server 160 to communicate with other servers and devices on the network, such as over an IP network. Examples of a network interface 570 include, but are not limited to, modems such as cable, ADSL, or optical, interfaces that communicate through wireless frequencies or infrared frequencies, and network interface cards. A non-limiting example of a network interface 570 includes a transmitter or means for transmitting read requests and write requests, a receiver or means for receiving tables, data structures, reports, and other information, cell phone, or WAN radio for communicating with one or more mesh gates 140.

[0159] The stack 580 is used to run applications by wrapping data packets in headers and trailers and later stripping the headers and trailers used by the layers of the stack 580. The headers and trailers adhere to the requirements of the various protocols used by the layers of the stack 580. Additional information regarding stack 580 can be found in U.S. patent application Ser. No. _____ (TR0003) filed contemporaneously herewith entitled "Efficient And Compact Transport Layer And Model For An Advanced Metering Infrastructure (AMI) Network," which is incorporated herein by reference in its entirety.

[0160] An AMI network should advantageously adhere to one or more standards applicable in the region in which it is operated. In at least one non-limiting embodiment, the AMI network in or with which aspects of the invention operate

adheres to the ANSI C12 suite of standards. Other embodiments may be configured to adhere to other regional or international standards or ISO standards as may be adopted from time to time. In particular, ANSI C12.19 defines and standardizes the contents and formatting of tables of data, and ANSI C12.22 establishes the protocol standards which allow transport of table data over a reliable network communications system. The ANSI C12 protocol is implemented at the application layer and is supported by the transport layer, the mesh layer, and the data link and physical layers established under the IEEE 802.15.4 protocol. Specific reference is made to the 1997 version of ANSI C12.19 and the 1999 versions of ANSI C12.18 and ANSI C21, all in effect as of 2006. In addition, reference is made to the 2008 version of ANSI C12.11 and the 2003 and 2006 versions of IEEE 802.15.4. Each of these standards is hereby incorporated by reference. It may be appreciated by those workers having ordinary skill in the art in light of the description provided here that the invention and embodiments, aspect, and features of the invention are not limited to specific standards, though some embodiments may apply specifically to such standards, and that the invention and aspects thereof are compatible with existing standards as well as enhancements, revisions, and future modifications of such standards. One example of another standard that may be used is the IEC standard.

[0161] Exemplary Types of Node Devices and Meters

[0162] A utility meter collects data about utility usage or consumption at residential and commercial customer sites. Examples of utilities from which usage data may be collected include electricity, gas, water, and communications services. Usage and/or consumption may represent total usage during a period of time, temporal consumption such as instantaneous usage of one or more utilities, total consumption during a defined time period, peak usage, usage during peak or off-peak times, or any other usage or consumption information independent of whether it is of an instantaneous, real-time, cumulative, pattern, or other usage.

[0163] In accordance with one non-limiting embodiment, meters are conveniently categorized as one of two basic categories of utility meters, dumb meters and smart meters. FIG. 6A shows a block diagram 600A of a node, also referred to as a node device, 121, 122, 123, 131, 132, 133 that includes a dumb meter 620. A dumb meter 620 has a sensor 610 or other instrumentation which counts or measures the usage of a utility such as electricity, gas, or water; a dumb meter 620 does not process any usage data, or at least does not process any usage data beyond that processing that might be characterized as a collection or capture processing. The data captured by a dumb meter sensor 610 may be stored and transmitted to a collector through the use of an external electronics circuit or radio, not considered part of the dumb meter, such as may be implemented as one or more printed circuit card or chips which may include a register 612 for storage and processing functions and a communications card or circuit 614 for communications functions. The register 612 processes measurements made by the sensor 610 and stores data structures in the form of lists or tables such as meter readings, profiles, and alarms data structure(s). The register 612 also creates ANSI C12.22 envelope (or other similar or equivalent proprietary, regional, international, or ISO standard or protocol envelope) around the data structures to create a report. The communications card or circuit 614 transmits the reports using the ANSI C12 protocols (or other protocol) to the collector for transmission to the head end server. The reports may

be either pulled by the collector and/or head end server or pushed by the communications card or circuit 614. A dumb meter 620 in combination with the afore described circuits or printed circuit card containing such circuits which perform the register 612 and communications 614 functions is referred to herein as a node or node device 630 in the advanced metering infrastructure (AMI) network. Reports may be either pulled or pushed from a node with a dumb meter. Pulling is always supported on demand. Push may also be extended to the system, and may for example be supported as an extension to the C12 standard.

[0164] FIG. 6B shows a block diagram 600B of a node, also referred to as a node device, 121, 122, 123, 131, 132, 133 that includes a smart meter 650. Similar to the dumb meter sensor 610, a smart meter 650 also has a sensor 640 or other instrumentation which counts or measures the usage of a utility such as electricity, gas, or water. However, in addition, the smart meter 650 includes a register 642 for processing measurements made by the sensor 640 and storing data structures in the form of lists or tables such as meter readings, profiles, and alarms data structure(s). An external electronics circuit or radio 644, not considered part of the smart meter 650, such as may be implemented as one or more printed circuit card or chips may implement communications with a radio, or a receiver 645 or means for receiving data or reports and a transmitter 645 or a means for transmitting data or reports, processing with a processor 646 or means for accessing data structures on lists stored in the register 642, and memory 647 or means for storing data. In one non-limiting embodiment, a smart meter 650 in combination with a communications card or circuits 644 is referred to as a node or node device 660 in the advanced metering infrastructure (AMI) and affiliated system and method.

[0165] Moreover, in at least one non-limiting embodiment, a node 121, 122, 123, 131, 132, 133 in the AMI may comprise an electronics card (or equivalent circuits) in combination with a utility-consumption or utility-measuring device including, but not limited to, meters on larger appliances such as a dishwasher or washing machine, meters on any device that consumes energy, or other utilities such as light bulbs, gas stoves or ovens, and/or thermostats. These devices may be either dumb, in which case the electronics card provides both register and communications functions, or smart, in which case the electronics card provides only communications functions.

[0166] A communications card 644 may retrieve data from a smart meter 650, by advantageously connecting to the meter using the ANSI C12.18 point-to-point connection protocols or an equivalent adopted standard for the region, such as an ISO or other international standard connection protocol.

[0167] FIG. 7 illustrates an example of a method 700 of downloading new data not previously downloaded from a smart meter register 642 to a communications card 644 at the node. In one exemplary but non-limiting embodiment, at block 710 a typical smart meter 650 stores in a register 642 four types of lists or tables: a history logger list or table, an event logger list or table, a profile list or table, and a self-read list or table. Other data structures may alternatively be used and some of these lists, tables, or other data structures may optionally but advantageously be used. The history logger list or table advantageously stores lists of events including, but not limited to, power quality such as low voltage, tampering, hardware diagnostics, and received load control commands. The event logger list or table advantageously stores events

that may have an impact on the utility bill, such as power outages, and is specified by the ANSI C12 protocol. The profile list or table advantageously captures different measurements at specific intervals of time. The self-read list or table advantageously captures all the information needed to bill a customer, such as by way of example but not limitation, the number of kilowatt-hours used at a certain tiered price and maximum demand.

[0168] At block 720, for each list or table stored by the smart meter register 642, the register also advantageously stores a corresponding sequence number. The sequence number rules and syntax may be as identified or defined by the ANSI C12.19 protocols or other regional or international protocols. At the installation of a meter, the sequence number of each list is set to zero or some other predetermined or dynamically determined reference or starting number. A sequence number is incremented by one (or other predetermined or dynamically determined increment) when a new piece of data is appended to the list. The sequence numbers are advantageously never reset during the life of the smart meter so that each sequence number is unique to the life of the meter and not reused.

[0169] Each time the communications card (or equivalent circuits) 644 downloads data from the smart meter register 642 to memory 647 on the communications card 644, the sequence numbers corresponding to the last downloaded data entry of each list are also downloaded and stored in memory on the communications card. Thus, at block 730 the communications card 644 uses its processor 646 to access only the data entries on each list of the register 642 having a sequence number greater than the sequence number stored previously in the memory 647 on the communications card 644 for that list. For example, the processor 646 may read the sequence number entries for each list and compare them to the sequence number previously stored for that list, and when the processor arrives at a sequence number greater than the previously stored sequence number, the processor will access the corresponding data entries.

[0170] Then at block 740, the communications card 644 uses its receiver 645 and memory 647 to download and store new data and their corresponding sequence numbers, where the new data has not previously been downloaded from the smart meter register 642.

[0171] Besides storing the sequence numbers for each list stored by the smart meter register 642, the communications card 644 also needs to know the table (or other data structure) identification number which identifies the type of table and the frequency at which the smart meter captures the data for that particular table. The information is stored in the smart meter register 642. At block 750, the communications card 644 downloads this information.

[0172] At block 760, a smart meter may also store data structures that are not list-based or table-based, for example, such data structures may include or store statistics, Global Positioning System (GPS) location of the meter, configuration information, or combinations of any two or more of these or other information.

[0173] In one embodiment, after the communications card 644 downloads data from the smart meter register 642, a processor on the communications card 644 creates an ANSI C12.22 envelope (or other similar or equivalent proprietary, regional, international, or ISO standard or protocol envelope) around the data to create a report. Reports may include, but are not limited to, identification information, status of the

node such as low battery, detection of tampering at the node, power failures, power quality, and leakage; configuration of the meter; data formatting information, utility meter readings, consumption statistics, and a profile of the interval period at which utility readings are stored. The node 660 may push the reports to the system head end server 160 for processing through a collector 140. Alternatively, the head end server 160 and/or the collector 140 may pull the reports from the node 660.

[0174] In one embodiment, after the communications card 644 downloads data from the smart meter register 642, the communications card 644 does not have the ability to create a report from the data. Thus, the collector and/or the head end server would pull the information from the communications card 644, rather than having the communications card push the data to the collector and/or the head end server. The processing of pulling the data from the communications card 644 follows conventional ANSI C12 Protocol Specification for Electric Metering (PSEM).

[0175] Installation of Nodes

[0176] Prior to joining an AMI network and transmitting data, a node must undergo an installation procedure as specified by ANSI C12 protocols (or other protocol) and briefly described here. FIG. 8 illustrates an example procedure 1800 for installing a new node.

[0177] In one non-limiting embodiment of an installation procedure, at block 1810 the node is initially provided the option of engaging in a commissioning communication with the commissioning server at the system head end server 160. The content of the commissioning communication depends on the type of device at the node, for example, a meter or a thermostat, the data structures supported by the node, and the utility-owner of the node. Thus, a thermostat may dump or upload its control schedule to the head end server 160 during the commissioning communication.

[0178] The next step in the installation procedure is association of the node with a new collector at block 1820. A mesh network is a wireless network configured to route data between nodes within a network. It allows for continuous connections and reconfigurations around broken or blocked paths by retransmitting messages from node to node until a destination is reached. Mesh networks differ from other networks in that the component parts can all connect to each other via multiple hops. Thus, mesh networks are self-healing: the network remains operational when a node or a connection fails. A mesh network may be used to implement any of the WAN 110, NAN 120, and HAN 130 networks within an AMI network 100. For convenience, a mesh network may be referred to here simply as either a network or sub-network.

[0179] FIG. 9 illustrates an example node to new-collector association procedure 900 for associating a node with a new collector. After association, the node may utilize the new collector and the new collector's sub-network to communicate sensor readings and receive instructions or commands.

[0180] At block 910, the node may use a receiver or radio means 265 to detect active networks within radio range and collecting information on each. For example, the node may determine a signal strength for a particular transmission frequency, a network name, a supported version number of each network. The active networks may be compiled in a list by a processor or microcontroller 245 in memory 255.

[0181] Each network may be associated with a collector and may periodically broadcast a banner or other information containing a network name and a network identifier. The node

may attempt to receive and parse or interpret the banner in order to determine nearby networks having a minimum signal strength. In one non-limiting example, only the collector may broadcast the banner, but each node within the network may forward the banner. In this way, nodes outside direct radio range of the collector may still participate in the network through nearby neighbors.

[0182] At block **920**, the node may use its microcontroller **245** to select a prospective collector with which to associate based upon the information acquired at block **910** about active networks. From the list of networks collected above, the node may parse the network name for a network prefix, network suffix, or other identifier. The network prefix may determine, in part, services offered by the network. Alternatively, the network prefix may determine a provider of the network. Based on the services offered or a registered provider of a network, the node may select the prospective network.

[0183] The node may further select the prospective collector based on a signal strength. A strong signal may be preferred over a weak signal. The node may further select the prospective collector based on a supported version number. It will be appreciated that the node may be outside direct radio range of the collector. However, communications may be forwarded through neighboring nodes in accordance with network protocols.

[0184] At block **930**, the node may transmit a neighbor request to the prospective collector selected above. The neighbor request may include a node identifier and relevant meter status and be transmitted to the collector associated with the network. For example, the meter status may include a list of sensors provided by the node and services requested by the node. A schedule of supported sensor reading transmission may be transmitted, for example, indicating whether the node will transmit sensor readings every minute, every hour, every day, or according to some other predetermined or dynamically determined schedule or the like, and any black-out periods.

[0185] As described above, if the node is not in direct radio range of the collector, the communications may be forwarded or relayed through the network. The node may first transmit to a neighboring node, which then forwards or relays the communication on. It will be appreciated that the above sub-process may be executed automatically on power-up or upon collector failure of the collector currently associated with the node. No further input from a user may be required.

[0186] At block **940**, the node may receive a neighbor response. Responsive to receiving the neighbor request from the node, the collector may compile or assemble and transmit the neighbor response via the network. The neighbor response may provide information to the node regarding associating with the prospective network.

[0187] For example, the neighbor response may include a next hop to the collector, a number of hops to the collector, a path quality, a collector load, a router load, or some combination of these. A next hop to the collector may describe the next node on the path to the collector. A path quality may be an indicator indicating a signal quality of the path to the collector. A collector load may indicate remaining capacity at the collector. A router load may indicate remaining capacity at a node next on the path to the collector, if the node is not in direct radio range with the collector and forwarding is required.

[0188] At block **950**, the node may transmit an association request to the collector via the network. The association request may indicate a node's desire to associate with the prospective network selected above. For example, the association request may further include a node identifier.

[0189] At block **960**, the node may test whether an association response was received. Responsive to receiving the association request from the node, the collector may associate the node with the network and transmit an association response. Similar to above, the association response may be forwarded through the network.

[0190] For example, the collector may check a collector load factor before allowing the node to associate. The collector may also authenticate the node before allowing the node to associate. For example, the node may transmit an authentication key verifying its identity. For example, the collector may look up the node table at a server or in a look up table or in another data structure to verify the node is authorized to associate.

[0191] At block **970**, the node may associate with the network; thus, the node may transmit reports and data to the head end server via the new collector and also receive messages from the head end server. The node may update an internal neighbor table with a network identifier, a network name, and neighbor information such as a next hop and a number of hops to the collector. In this context a hop is a transmission of a message from one node to another node until a destination, the collector, is reached.

[0192] Future communications may be transmitted to the next hop, a nearby neighboring node in the network. After the node is associated with the network, it may act as a neighboring node for another new node searching for a network to associate with.

[0193] It will be appreciated that in the absence of an associated network, the node may continue to store sensor readings for future transmission.

[0194] Following the association of the node with a new collector at block **1810**, the next step in the installation of a node is registration according to an applicable protocol specification, such as for example an ANSI C12.22 registration according to the ANSI C12.22 protocol specification. The node transmits registration information in a simple ANSI C12.22 registration message to the collector, and the collector transmits the information to the master relay which includes a routing table for tracking all of the C12.22 nodes registered to a C12.22 network service provider.

[0195] After the registration step, at block **1830** of the example procedure **1800** for installing a new node, the master relay notifies the head end server **160** that there is a new node **121, 122, 123, 131, 132, 133** on the network **100** and that the node has been registered according to the applicable protocol specification such as according to the ANSI C12.22 protocol specification.

[0196] Finally, at block **1840** the installation procedure concludes with an initial communication from the server to the node, requesting static information. Static information encompasses information related to the node which does not change and thus does not need to be transmitted to the head end server during future data transmissions. Static information may include, but is not limited to, the type of measurements that the meter at the node will make, the frequency at which the meter will make the measurements, the identification number of the meter, and/or any information that may be located on the nameplate of the meter.

[0197] The installation procedure serves to inform the network of the presence of the node, and the data collection process may proceed upon successful completion of the installation process.

[0198] Transferring Data from the Node to the Collector

[0199] When reports or data are transferred from the node to the collector through a pull process, the collector must know what type of information to retrieve and the type of meter from which the information is being retrieved. The node merely operates as a server of data acquired at its location. The collector runs a specific application program to pull information from a specific kind of meter and to read the specific data structures used by the meter.

[0200] FIG. 10 illustrates a first example Dumb Meter Node-to-Collector data pull procedure 1000 for transferring data from a node to a collector using a first pull scenario. In the first scenario, the collector 140 pulls a report from a node 630 with a dumb meter 620. The collector 140 uses a radio 355, a transmitter, or other transmission means to transmit a pull request at block 1010 to the node 630. At block 1020, a report is created at the register 612 at the node 630. At block 1030, the communications card 614 transmits the report to the collector 140 using a radio 265, a transmitter, or other transmission means.

[0201] FIG. 11 illustrates a second example Smart Meter Node-to-Collector data pull procedure 1100 for transferring data from a node to a collector using a second pull scenario in which the collector 140 pulls a report from a node 660 with a smart meter 650, where the node 660 is capable of creating a report. The collector 140 transmits a pull request at block 1110 to the node 660. At block 1120, the communications card 644 downloads the data from the register 642 of the smart meter 650 as detailed above by procedure 700 in FIG. 7. At block 1130, the communications card 644 creates the report from the data. The communications card 644 then transmits the report to the collector 140 at block 1140.

[0202] FIG. 12 illustrates a third example Alternative Smart Meter Node-to-Collector data pull procedure 1200 for transferring data from a node to a collector using a third pull scenario that also involves a smart meter, but where the smart meter does not have the ability to create a report so the device-agnostic collector pulls the data and creates the report for the meter. The collector 140 transmits a pull request at block 1210 to the node 660. At block 1220, the data structures from the register 642 in the smart meter 650 are downloaded to the communications card 644. At block 1230, the communications card 644 transmits the data structures to the collector 140. The CPU 345 of the collector 140 then creates the report from the data structures at block 1240, where a report includes an envelope or container for the data structures, such as an ANSI C12.22 envelope around the data structures.

[0203] Reports created by the collector or pulled by the collector from the node are subsequently transmitted to the head end server by the collector. Alternatively, the head end server may send a pull data command to the collector, rather than having the collector initiate a pull request. In this case, the collector could either pull data from all nodes from its sub-network or could pull data from a particular node specified by the head end server. One specific example in which data is pulled by the head end server is an on-demand response, where only data is sent rather than an entire report. An exemplary process by which the collector transmits the reports to the head end server is described in more detail

below in the section entitled, "Three Step Synchronization Between Head End Server and Collector."

[0204] In contrast to a pull process, a push process occurs when a node initiates transmission of reports or data to the collector and/or the head end server. Recall that a report may consist of data plus an envelope for that data. The envelope may include such information as security information, sequence information, address information, etc. A node may be configured to actively initiate transmission of data or a report to a collector and/or head end server rather than merely responding to a pull request for information. The collector that receives data or reports pushed from a node should be capable of receiving unsolicited transmissions initiated by the node, regardless of the type of meter at the node or the type of data structures transmitted by the node. Consequently, this type of collector may be termed a meter-agnostic, data structure-agnostic, or device-agnostic collector. Because some nodes may not have the ability to push information, a device-agnostic collector may also advantageously need to be capable of both pulling data or reports from one node as well as receiving pushed data or reports from another node. More details on the push process are provided below in the section entitled, "Pushing Reports from the Node to the Collector." These various agnostic capabilities are referred to by the general phrase device-agnostic.

[0205] One of the functions of at least some mesh gates may be the collector function. Every valid message read is stored in a First-In-First-Out (FIFO) buffer (or other buffer as are known in the art) The head end server 160 has a protocol for head end synchronization with a collector. There is provided an address list, and the head end sends the actual sequence number for each address, so that based on the sequence number the mesh gate will only transfer new reports.

[0206] FIG. 13 is a flow chart illustrating an example 1300 of a method of pushing data to a collector from a node, according to an aspect of the invention. A node may push information to a device-agnostic collector according to various policies or criteria.

[0207] At block 1310, a node device collects data through an input device. For example, profiler data monitors consumption of a utility, such as for example, voltage drop or variation over time due to electricity usage or quantity (e.g. cubic feet) of water consumed. The profiler data is measured and collected at regular intervals, such as every hour.

[0208] After the data is collected, at block 1320, the data is stored in memory. For the profiler data example, the profiler data is dumped or stored into a block of memory. A block is the repository for data collected at known, regular intervals. The block is an efficiency mechanism which does not require a time stamp to be included at each data capture because each time data is sent to the block is known.

[0209] At block 1330, a report is created by the node's processor from the data. For the profiler data example above, when the block is full, a report is created, and the report is appended to the node's reporting list.

[0210] At block 1340, the node transmits the reports on the reporting list to the collector at the next scheduled reporting time window by using a radio, transmitter, or other transmission means.

[0211] In the profiler data example above, the node pushed information to the collector when the information was created. A node may also push information on a schedule such as the accumulation of statistics relating to energy usage. At appropriate pre-programmed intervals, the node transmits the

data packet to the device-agnostic collector. The device-agnostic collector receives the data packet and forwards the data packet to the head end server. Additional description is provided herein elsewhere relative to particular batch reporting and real time reporting.

[0212] A device-agnostic collector does not need any knowledge about the particular type of meter or utility-consumption device at the node pushing the data and reports nor the particular data structure used by the node. Consequently, new metering devices capable of pushing information may be installed onto a network, but the device-agnostic collector does not need to upgrade its database to include information about the new type of meter. As discussed in more detail below in the section "Pushing Reports from the Node to the Collector," the node addresses data packets to be processed in a particular manner to certain pre-assigned collector ports, and the collector processes the data packets according to the port that the packet arrives at. In contrast, if a collector were required to pull information from a new type of metering device, the collector would need a specific driver or other specific application for each type of meter and update the driver for each new revision, adding complexity to the collector. The device-agnostic collector also does not need to understand any special data structures pushed by a node because the collector does not read the contents of the data packet, only the header on the data packet. A node capable of initiating a transmission will push the data using conventional data structures as established by the applicable protocol implemented, such as the ANSI C12 protocol. A report is created when a package, container, or envelope are created around the data structure, such as for example, when the ANSI C12 envelope is created around the data structure. Reports that can be pushed by a node may include, but are not limited to, identification information, status of the node such as low battery, detection of tampering at the node, power failures, power quality, and leakage; configuration of the meter; data formatting information, utility meter readings, consumption statistics, and a profile of the interval period at which utility readings are stored. One or more of these reports are captured at the appropriate interval for that particular data structure and then appended to a reporting list stored at the node. The processor, microcontroller **245**, or other means creates a reporting list in memory **225**, and newly created reports are appended by the processor, microcontroller **245**, or other means to the reporting list. The reporting list is then transmitted by the node one report at a time, addressed to the appropriate device-agnostic collector port at preprogrammed intervals and/or according to other policies or rules. Thus, the processor or microcontroller **245** uses the reporting list to keep track of the new reports that need to be transmitted to the collector.

[0213] In accordance with one non-limiting embodiment, for security purposes, the data structures in the data packet from the node are advantageously encrypted with a data key which the device-agnostic collector is not able to decrypt. Only the head end server is able to decrypt the data from the nodes. Thus, the manner in which the device-agnostic collector responds to a data packet, whether received from a node or from the head end server, may depend upon which port of the device-agnostic collector the data packet is addressed to rather than the content of the data packet.

[0214] The device-agnostic collector's ports may be reserved according to the type of response expected from the collector including, but not limited to, forwarding the data

packet from the node to the head end server without processing the data packet, capturing and aggregating the data prior to forwarding batched data from multiple nodes to the head end server, and forwarding registration information from the node to the head end server. Thus, a node pushing data or a report in a data packet to a device-agnostic collector must address the data packet to the appropriate pre-assigned port of the collector.

[0215] The pulling or pushing of data advantageously occurs at the application layer level in the AMI network. As established by the IEEE 802.15.4 protocols, the application layer is supported by the underlying layers, the transport layer, the data link layer, and the physical layer. Layer boundaries are somewhat flexible and elastic so that the data pushing or pulling may not strictly be limited to a particular layer.

[0216] Collision Avoidance-Transmitting Packets from Node to Collector

[0217] In an AMI network **100**, a collector **140** may be required to service up to and more than thousands of different nodes **121, 122, 123, 131, 132, 133**. Nodes in any particular sub-network **120, 130** may be programmed to record meter data at the same time, for example every hour on the hour. If data is pulled from the nodes on a collector's sub-network by the collector **140** or head end server **160**, collision of data on the NAN **120** is minimized because each node responds with data packets **600A** recorded at the hour only when queried by the collector **140**. However, if the nodes in a sub-network push the recorded data to a collector during a pre-scheduled time window, thousands of data packets may be transmitted simultaneously, resulting in multiple collisions. Although the IEEE 802.15.4 MAC layer implements carrier sense multiple access (CSMA) for handling collisions, the MAC layer is only capable of typically handling no more than approximately five nodes transmitting simultaneously.

[0218] FIG. **14** shows an example of a block diagram **1400** of a node communicating with a collector through a network, according to an aspect of the invention. The node **1410** includes a radio **1420** and a microcontroller **1430**. The radio includes a transmitter or other means for transmitting data packets and a receiver or other means for receiving an acknowledgment from the collector **1450**. The microcontroller **1430** has a software module **1440**, substantially random time slot generator, or other means for generating one or more random or substantially random time slots within a pre-scheduled reporting time window. The microcontroller **1430** also has another software module **1442** or other means for seeking a new collector.

[0219] The node **1410** transmits data packets to a collector **1450** through a network **1470**. The collector includes a radio **1460** or other receiving means for receiving the data packets received from the node **1410** and transmission means for transmitting and acknowledgment after receipt of each data packet from the nodes.

[0220] FIG. **15** illustrates an example solution **1500** for handling a situation where even staggered transmission times for each node may still result in collisions. At block **1510**, a random or substantially random time slot within a pre-scheduled reporting time window is generated for each of the nodes. While the generation of random time slots within a reporting time window for each node would be ideal in minimizing the collision of transmissions from the nodes to the collector, the assignment of substantially random time slots would suffice provided the substantially random slots do not permit more than approximately five nodes to transmit simul-

taneously. These time slots within the prescheduled reporting time window are referred to as checkpoints for the nodes. At block 1520, each node transmits its reports as data packets to the collector 140 during its checkpoint. Each node uses a radio 265, transmitter, or transmission means to transmit the data packets. The result is that data packets sent from the different nodes to the collector are transmitted in a temporally staggered manner over prescheduled reporting time window rather than being transmitted simultaneously.

[0221] Checkpoints may be generated by nodes and there is also an option for a mesh gate to trigger the generation of a checkpoint. Using a mesh gate to trigger checkpoint generation may be advantageous because the mesh gate may typically be in a better position to control collisions, but may sometimes have the somewhat disadvantageous effect of creating more message traffic. Either or both checkpoint generation schemes may be utilized depending upon the tradeoffs. Furthermore, if a tree is broken, the node needs to initiate the checkpoint generation because under these conditions, the mesh gate cannot read back to the node. In one non-limiting example, the node initialization of a checkpoint serves as a backup update routing information for the mesh gate. A node can normally always find a mesh gate but the mesh gate may not always find a node.

[0222] At block 1530, the collector transmits an acknowledgement to each node if the collector received a data packet from that node. The collector uses a radio 355, receiver, or receiver means for receiving data packets and a radio 355, transmitter, or transmission means to transmit the acknowledgement.

[0223] At decision block 1540, each node uses a radio, receiver, or receiver means for checking whether it has received an acknowledgement from the collector if it sent a data packet. If each node that sent a data packet to the collector received an acknowledgement (block 1540—Yes), the process ends at block 1570. The data packet may not reach the collector, or an acknowledgement may not be received by the transmitting node for a variety of reasons including, but not limited to, interference at the RF frequency of transmission and collisions with another packet. If the data packet does not arrive at its destination, as indicated by the lack of an acknowledgment from the collector, the transport and MAC layers may attempt to retransmit the packet. If the transport and MAC layers are not successful in re-transmitting the packet, in one embodiment, the node may be configured to re-transmit the packet to the collector again before the next scheduled reporting time window. Thus, if at least one node did not receive an acknowledgement from the collector after transmitting a data packet (block 1540—No), at block 1550 a second random or substantially random time slot within a pre-scheduled re-try time window may be assigned to the unacknowledged nodes. The re-try time window may or may not be the same as the prescheduled reporting time window used at block 1510 above. At block 1560, each unacknowledged node transmits its report to the collector again during its second assigned checkpoint. The process ends at block 1570.

[0224] In one embodiment, if there are nodes that did not receive an acknowledgement, these nodes may be configured to wait until the next scheduled reporting time window before re-transmitting the packet.

[0225] Any algorithm which implements an effective randomization or pseudo-randomization of transmission times

for the nodes in a collector's sub-network is suitable. One example of a randomization algorithm or procedure is presented here.

[0226] Under the IEEE 802.15.4 protocol, 16-bit addresses are assigned to each meter sequentially by the mesh gate or coordinator. As described in the IEEE 802.15.4 protocol, a coordinator implements a general model of communication which allows it to talk to any other device and also relays messages. The mesh gate 140 performs the functions of the coordinator in the example network topology in FIG. 1.

[0227] Bits one through seven of the address are used to represent bits seven through thirteen of the generated random number. Bits one through seven of the meter manufacturer's serial number and bits one through seven of the actual meter reading are used in computing bits one through seven of the generated random number. Bits one through six of the generated random number are obtained by performing an exclusive OR operation on the first six bits of the meter's serial number and the first six bits of the actual meter reading. Bit seven is obtained by performing an exclusive OR operation on the seventh bit of the meter's serial number, the seventh bit of the actual meter reading, and bit one of the 16-bit address. Bits eight through thirteen correspond respectively to bits two through seven of the 16-bit address. The generated random number has thirteen bits; thus it represents a number between 0 and 8191. This generated random number is multiplied by the total time window allowed for the transmission of all the nodes in a collector's sub-network and then divided by 8191 for scaling. The resulting value is the time during the window at which a node is assigned to transmit its data packet during the time window.

[0228] Alternatively, a random or substantially random time slot or number generator may be used to generate the times at which each node is permitted to initiate transmissions to the collector, and the time slot or number generator may be performed by a processor or microcontroller 245 running an algorithm.

[0229] Last-Transferred-Pointer

[0230] A NAN-associated node 121, 122, 123 may lose communications with its servicing collector 140 for a variety of reasons including network outages. However, each of the reports stored at the node on the reporting list remain in queue on the reporting list in flash memory, even during a network or power outage. The reporting list is a circular buffer that advantageously uses a first-in first-out protocol. Typically, the reporting list can store approximately one year of data, but different size buffers may be used to store different durations or quantities of data. A person skilled in the art will understand that an arbitrary amount of data may be stored on the reporting list.

[0231] The node implements a last-transferred-pointer in conjunction with the reporting list. FIG. 16 illustrates an example procedure 1600 for using the last-transferred-pointer. The last-transferred-pointer is a programming language data type stored in memory 255 whose value points to or flags the address in memory 255 of the last report on the reporting list that was acknowledged as having been received by the collector. The report pointed to by the last-transferred-pointer may be referred to as the flagged report. The report to be transmitted next is the one immediately following the flagged report.

[0232] At block 1602, a node creates one or more reports using a processor. At block 1604, the node appends the one or more reports to a reporting list in a memory.

[0233] At block 1610, at a node's assigned checkpoint, the node uses a radio 265, transmitter, or transmitter means to transmit one report in the form of a data packet from the reporting list to the collector. If the collector receives the data packet, it will send an acknowledgment to the node.

[0234] The node uses a radio 265, receiver, or receiver means to check at decision block 1620 for an acknowledgment generated and sent by the CPU 345 of the collector. If a report transmission is not acknowledged (block 1620—No), the node does not increment the position of the last-transferred-pointer and re-transmits the same report at the appropriate re-transmission time at block 1610. If the report is acknowledged (block 1620—Yes), at block 1630 the node updates the last-transferred-pointer by using a processor or microcontroller 245 to increment the last-transferred-pointer stored in memory 255 to point to the report just transmitted and acknowledged. At decision block 1640, the node decides whether there are more reports to transmit to the collector. If the node has more reports to transmit to the collector (block 1640—Yes), it starts the process from block 1610. Otherwise (block 1640—No), the process ends at block 1650.

[0235] There are two situations which will cause a node to not receive an acknowledgement. In the first situation, the report transmitted by the node is not received by the collector. When an acknowledgement is not received, the transport and MAC layers first attempt to re-transmit the report. If the transport and MAC layers are not successful in re-transmitting the report, the node may be configured to re-transmit the report to the collector again before the next scheduled checkpoint. Alternatively, the node may be configured to wait until the next checkpoint before re-transmitting the report. In either case, the node will know which report to re-transmit because it is the report immediately following the one that the last-transferred-pointer is pointing to on the reporting list. Using the last-transferred-pointer scheme, the node automatically knows which reports have been acknowledged and do not need to be re-transmitted.

[0236] In the second situation, a report transmitted by the node was actually received by the collector, but the acknowledgement from the collector was not received by the node. Because the collector actually did receive the transmission, the collector's memory buffer has already been updated with the report from the node. However, if an acknowledgement is not received by the node, the transport and MAC layers first attempt to re-transmit the report. If the transport and MAC layers are not successful in re-transmitting the report, the node may be configured to re-transmit the report to the collector again before the next scheduled checkpoint. Alternatively, the node may be configured to wait until the next checkpoint before re-transmitting the report. In either case, the node will re-transmit the same report because it is the report immediately following the report pointed to by the last-transferred-pointer. When the collector receives the new transmission, it will re-acknowledge the receipt of the report but will discard the duplicate report which has been sent twice by the node. Upon receipt of the acknowledgement, the node will update the last-transferred-pointer to point to the last report acknowledged by the collector.

[0237] In a network which implements data pulling, should a collector fail permanently for any reason such as software or hardware failure, the system head end server should direct another collector to take over for the failed collector. However, in a network in which nodes push data and reports to the collector, the nodes are autonomous and will seek another

collector and its associated sub-network on its own and start transmitting reports and/or data to the new collector as discussed above. No reports will be lost due to the failed collector because a node will transmit its reporting list starting from the next report after the one flagged by the last-transferred-pointer.

[0238] Pushing Reports from the Node to the Collector

[0239] There are multiple ways for a NAN-associated node 121, 122, 123 to push reports to the head end server 160 through the collector 140. Examples of non-limiting ways to push reports include, but are not limited to, (i) batch reporting, and (ii) real-time reporting. In accordance with one non-limiting embodiment, most reports are sent through batch reporting, for example, interval data and work files. The reports which are batched are typically reports not required urgently so they are batched for efficiency. Batch reporting makes maximum use of the data packets sent across the network by minimizing the amount of traffic transmitted. Real-time reporting is advantageously reserved for data which may be needed urgently by the head end server, for example, a meter tamper alert, power outage management events, and feedback from network load control devices like switches.

[0240] With both methods of reporting, NAN-associated nodes 121, 122, 123 first transmit data packets to the collector 140. The node alerts the collector as to the type of report being sent by addressing a particular port of the collector in the header of the data packet. One non-limiting embodiment of the format of a data packet 1700A, including header formatting, is shown in FIG. 17. The packet 1700A may be composed of an IEEE 802.15.4 MAC header 1710, a layer header such as a Trilliant (Trilliant Networks, Inc., 1100 Island Drive, Redwood City, Calif. 94065) mesh layer header 1712, a transport header such as a Trilliant mesh transport (TIP) header 1714, an application layer header 1716, and an application payload 1718 such as a report. Each of the headers for the individual layers is used by the respective stack layer for transmitting the packet to its destination.

[0241] The collector port addressed by the node when transmitting a report resides in the transport TIP header 1714. When the transport header is the TIP transport header, the TIP header 1714 contains the fields 1700B shown in FIG. 17 which includes flags 1720, a packet sequence number 1722, a last received sequence number 1724, and an extension field 1726. In one non-limiting embodiment, the extension field 1726 is an eight-bit field. Bits 7 through 5 are used to indicate the type of extension, for example TIP communication or intra-TIP communication.

[0242] When the extension type is set to "intra TIP communication", the extension field structure 1700C shown in FIG. 17 is used, including a source port number field 1730 and a destination port number field 1732. The eight bits of the extension field 1726 is divided between the source port number comprising bits 7 through 4 and the destination number comprising bits 0 through 3. When the node transmits a report to the collector, it is the source port number field which contains the particular collector port being addressed. The relevant details of the header which the collector needs to know, such as the addressed port, are made available to the collector by the operating system.

[0243] FIG. 18 shows an example of a block diagram 1800 of a system for transmitting a report from a node to a collector and from the collector to a head end server, according to an aspect of the invention. The node 1810 includes a transmitter

or radio **1820** or other means for transmitting a data packet to the collector **1830** and re-transmitting an unacknowledged data packet to the collector **1830** through the NAN network **1870**.

[0244] The collector **1830** includes a radio **1840** and a memory **1850**. The radio **1840** includes a transmitter or other means for transmitting data packets and a receiver or other means for receiving data packets. The memory **1850** includes source code having pre-assigned collector port assignments.

[0245] The head end server **1890** includes a radio **1895**. The radio **1895** includes a transmitter or other means for transmitting requests or information, such as a read request for a table of nodes serviced by a collector, a table of new reports, and/or a table of lost reports and a write request for a table of last sequence numbers from the collector. The radio **1895** also includes a receiver or other means for receiving tables and data packets through the WAN network **1880**.

[0246] FIG. **19** shows an example procedure **1900** for addressing reports to particular collector destination ports based upon the type of report. At block **1910**, specific collector ports are pre-assigned to correspond to high priority real-time reporting, aggregation for batch reporting, or other processing including, but not limited to, relaying an on-demand response. The pre-assigning of destination ports may be done by using source code stored in the collector's memory that has embedded within it the destination port assignments.

[0247] At block **1915**, the node collects and stores data for transmission to the collector and creates an envelope around the data to form a report and data packet.

[0248] At block **1920**, the node addresses a data packet containing one or more reports to the appropriate port of the collector, where the destination port is included in a data packet header. At block **1930**, the node transmits the addressed data packet to the collector.

[0249] At block **1940**, the collector receives the data packet and processes the data packet according to pre-assigned procedures corresponding to the particular collector port. At block **1950**, the collector transmits the data packet to the head end server, and at block **1960**, the head end server receives the data packet from the collector.

[0250] Thus, the collector does not need to read the payload of a data packet, only the necessary header information attached to the data packet. And in fact, the collector will not be able to read the payload of the data packets because the node encrypts the data with a data key, and only the head end server has the key to decrypt the data.

[0251] With real-time reporting, the collector CPU **345** initiates the transfer of information to the head end server. All of the timing intervals involved are known so it is possible to predict the time it will take for the information to arrive at the head end server. Typically, transmission of real-time data will take approximately five minutes or less but it may be appreciated that this duration is a typical value for a typical network and not a limitation of any or all embodiments.

[0252] With batch reporting, the collector CPU **345** or other processor or processing logic aggregates batch data from the nodes and waits for the head end server to request the data. The head end server may request the batch data once per day or once per hour, or according to other policies or rules and depending upon the system parameters and configuration. Thus, in at least one non-limiting embodiment, the time at which the data will arrive at the head end server is controlled by the head end server, not by either the node or the collector. A three-step synchronization process **2100** followed by the

collector **140** for transmitting batch reports to the head end server is detailed in a section entitled, "Three-Step Synchronization Between Head End Server and Collector" below.

[0253] Real-Time Reporting by the Collector

[0254] FIG. **20** illustrates an example collector relative reporting time determination procedure **2000** used by a collector to determine when to transmit real-time reports received from nodes to the head end server. At decision block **2010**, the collector determines whether a data packet **600A** has been received at a port assigned to receive real-time reports from the NAN-associated nodes **121**, **122**, **123**. If no new real-time reports have been received at the collector (block **2010**—No), the collector continues waiting for a data packet to be addressed to its real-time reporting port at decision block **2010**.

[0255] If the collector has received a data packet intended to be delivered to the head end server as a real-time report (block **2010**—Yes), at block **2020** the collector may be configured to delay the transmission of the report in order to aggregate more real-time reports. The delay, minimum delay (MIN_DELAY), is used to allow the network bandwidth to be used more efficiently. For example, consider the case where a particular node "node A" sends a real-time report to the collector, and another particular node "node B" sends a real-time report to the same collector some period of time later, for example 20 seconds later. It would be a more efficient use of network bandwidth for the collector to wait for a limited duration to aggregate both reports and send them together to the head end server rather than sending node A's report immediately and then sending node B's report 20 seconds later. The impact on the system of waiting a limited duration is negligible because if an alarm needs to be sent by the head end server based upon the two real-time reports, a 20-second (or other specified short) delay is not very critical.

[0256] Similar to the situation where reports sent by multiple nodes to a single collector may result in collisions, the reports sent by multiple collectors to the head end server also have the potential to result in collisions. Thus, after waiting for a period of time of minimum delay (e.g., MIN_DELAY) such as may apply to real-time event notification, power outage notification or the like, at block **2030** the collector will compute another delay having a substantially random duration equal to or less than a maximum random delay period (e.g., RND_PERIOD).

[0257] At block **2040**, the collector waits the duration of the calculated substantially random delay prior to sending any aggregated real-time reports. At block **2050**, the collector transmits the real-time reports to the head end server.

[0258] The substantially random delays computed by each collector will effectively result in a temporal staggering of reports from the collectors to the head end server over an extended period of time rather than multiple simultaneously transmitted reports. Multiple head end servers may use the same WAN so collisions may occur between transmissions from collectors to a first head end server and transmissions from the same or other collectors to a second head end server.

[0259] At decision block **2060**, the collector determines whether an acknowledgment has been received from the head end server. If the aggregated real-time reporting data packets from the collector does not reach the head end server or if no acknowledgement is received, a sequence of re-try attempts are made at the TCP/IP layer and at the application layer. If the re-try attempts made at the TCP/IP layer and application layer are also unsuccessful such that no acknowledgment is

received at the collector (block 2060—No), at decision block 2070 the collector determines whether the maximum number of re-try attempts established by the system has been reached. The system uses a number, maximum retry (MAX_RETRY), which caps or limits the total number of times a collector is permitted to re-send real-time reports if previous attempts have been unsuccessful. The cap is needed or at least advantageous to prevent a collector from constantly re-sending the reports and tying up network resources.

[0260] If the maximum number of re-try attempts have not been made (block 2070—No), the collector will return to block 2030 to compute a new substantially random delay to wait before re-transmitting the reports, as before. A similar algorithm is used as for generating the first substantially random delay, but a shorter scaling period is used to generate shorter delays for re-tries. Shorter delays are used because the majority of real-time reports sent the first time by collectors would have been successfully received and acknowledged, and there are far fewer collectors trying to re-send reports.

[0261] If the maximum number of re-try attempts have been made (block 2070—Yes), the collector returns to decision block 2010 to await the receipt of new real-time reports from the nodes. The collector does not re-send the aggregated real-time reports. In one non-limiting example, the collector may throw away or discard real-time reports that have been unsuccessfully sent. In another non-limiting example, the collector may save the real-time reports that have been unsuccessfully sent in order to send them in a next transmission of real-time reports. In accordance with one non-limiting embodiment, the reports may be lost only if there is a loss of power without a back-up power to preserve the reports. For example, if it is battery powered, the mesh gate may enter a power saving state, but before it shuts down or enters the power saving state it can store the reports for later retrieval and transmission. In non-battery implementations, such as those using a super capacitor, there may not be sufficient stored energy for the storage of such real-time reports to have sufficient priority to be stored before energy is lost. Note that non-limiting examples of battery powered nodes, such as battery powered mesh gates, have rechargeable batteries that may have battery capacity to operate 8 hours or more.

[0262] If the collector receives an acknowledgment from the head end server for the transmitted real-time reports (block 2060—Yes), at block 2080 the collector must wait for another period of time, sometimes referred to as an exclusion period. The exclusion period is a minimum period of time a collector must wait before starting a new communication in order to prevent the collector from continually transmitting packets over the network. This mechanism places a limit on the maximum number of connections the collector may make with the head end server per day. Thus, if the collector is constantly receiving alarms, it will be prevented from tying up network resources by trying to communicate with the head end server every time.

[0263] Synchronization Between Head End Server and Collector

[0264] The description below of a non-limiting example of a three-step synchronization process 2100 (or process for synchronizing) followed by the collector 140 for transmitting real-time reports to the head end server. In the described example, the synchronization process advantageously occurs at the application layer. However, the synchronization process 2100 is a communications protocol for use between the head end server and the collector, independent of whether a

layered structure is used, or when and if used, the particular layer at which it occurs in the layer structure. In one embodiment, the synchronization process 2100 may be used for communications between the head end server and the collector without the use of any layer structure.

[0265] It will also be appreciated that use of a layered structure in computer programs, devices, and computer network systems is common and that many protocols are layered, at least at a minimum of a physical layer and another layer that might or would include the other non-physical layer.

[0266] It will also be apparent to a person skilled in the art in light of the description provided here that the application layer is supported by the transport layer, the mesh layer, and the data link and physical layers established under the IEEE 802.15.4 protocol. The mechanics of the lower support layers which may readily be appreciated by workers having ordinary skill in the art in light of the description provided here will not be described here to avoid obscuring features and aspects of the invention.

[0267] The head end server may wait for the collector to send batch reports at a pre-assigned time window. Alternatively, the head end server may initiate a request for the delivery of the collector's aggregated batch reports, whether the reports were obtained by pulling or pushing, in a three-step synchronization process 1400. The three-step synchronization process contains the following three steps: (1) a first read request with a corresponding response to the read request, (2) a write request with a corresponding response to the write request, and (3) a second read request with a corresponding second response to the second read request. A read request made by a first device is the equivalent of retrieving information from a target device to the first device. A write request made by a first device is the equivalent of sending information from the first device to a target device. FIG. 21 shows the three-step synchronization procedure 2100. The details are described below.

[0268] First, at block 2110 the head end server uses a transmitter to send a read table request to the collector. Tables are the data structures used by the nodes. The specific table requested by the head end server in this first request is the list of ApTitles in the collector's database. ApTitle, defined in the ANSI C12 protocols, is the application process title of a node and corresponds to the address of the node. The ApTitle of a node within a particular collector's sub-network uniquely identifies that node.

[0269] At block 2120, the collector responds to the head end server's request by transmitting a table containing the ApTitles of all of the nodes in its sub-network. A processor 345 may apply one or more suitable compression algorithms to compress the ApTitles table transmitted by the collector. It will be apparent to a person skilled in the art in light of the description provided here that many different types of compression algorithms may be used. Non-limiting examples of compression algorithms include, but are not limited to, Burrows-Wheeler transform, dynamic Markov compression, and Huffman coding.

[0270] At block 2130, the head end server sends a write request containing a table of the last sequence number of each ApTitle in the table sent by the collector. The last sequence number for each node uniquely corresponds to the number of the last report received by the head end server from that node. At the installation of a meter, the last sequence number is set to zero. The last sequence number is incremented by one each time a new report is transmitted by a node, and the last

sequence number is never reset during the life of the meter. When a node transmits information to the collector, it does not have to transmit a complete table, even a transmission incorporating a partially written table where information has only been entered in specific fields of the table may constitute a report and be assigned a corresponding last sequence number.

[0271] Based upon the last sequence number of each node sent by the head end server, the collector will be able to identify the new reports that the head end server has not received since the last synchronization procedure. For example, if the last sequence number for node A in the head end server's database is 100, the collector knows that all reports having a last sequence number greater than 100 from node A need to be sent to the head end server.

[0272] At decision block 2140, it is determined whether the write request is proper. If the collector CPU 345 determines that the head end server's write request is not made properly (block 2140—No), the collector sends an error message at block 2145, and the synchronization process ends at block 2180. Possible reasons for generating an error message include lack of permission for the head end server to access the requested data and invalid table number provided by the head end server or that the head end server has not been authenticated for any reason. The collector may store in memory 335 lists of authenticated head end servers or lists of head end servers permitted to access certain types of reports from specific nodes, and the CPU 345 compares the requesting head end server identification with the lists stored in memory to determine whether the write request was proper.

[0273] If the collector CPU 345 determines that the head end server's write request is made properly (block 2140—Yes), the collector CPU 345 accepts the write request at block 2150.

[0274] At block 2160, the head end server sends a read table request to the collector for the new reports. The collector responds at block 2170 by fetching the appropriate reports the head end server has not received, optionally compressing them with suitable compression algorithms performed by a processor, and transmitting them to the head end server which receives the reports. Non-limiting examples of compression algorithms include, but are not limited to, Burrows-Wheeler transform, dynamic Markov compression, and Huffman coding. At this point, the transfer and synchronization process is complete at block 2180.

[0275] The collector uses a buffer for storing reports received from every node it services in its sub-network. The buffer may advantageously be a circular or cyclic buffer. A first-in first-out protocol is employed with the buffer, and may advantageously be sized to hold an appropriate quantity or time duration of data. In one non-limiting embodiment, the buffer is sized to hold or store approximately three months' worth of reports. It will be apparent to a person skilled in the art in light of the description provided here that the collector's buffer may store any amount of information. Consequently, the three-step synchronization process may be useful for recovering from a partial loss of data at the head end server's database by retrieving the lost data and/or lost reports from the collector.

[0276] FIG. 22 is a flow chart illustrating an example of a method 2200 of recovering lost reports at a head end server, according to an aspect of the invention. For example, if the head end server approximates that the most recent five days of reports received from a collector has been lost, the head end

server may employ the three-step synchronization process and enter the last sequence numbers for the nodes corresponding to a week prior to the database loss in order to retrieve all of the reports that the collector transmitted to the head end server within the past week. Blocks 2210 and 2220 correspond to blocks 2110 and 2120 of FIG. 21 described above. At block 2230, the head end server sends a write request containing a table of the last sequence number for each ApTitle, where the last sequence number corresponds to the last report received from the corresponding node or any report for that node that the head end server still has stored in its database.

[0277] At decision block 2240, it is determined whether the write request is proper. If the collector CPU 345 determines that the head end server's write request is not made properly (block 2240—No), the collector sends an error message at block 2245, and the synchronization process ends at block 2280. If the collector CPU 345 determines that the head end server's write request is made properly (block 2240—Yes), the collector CPU 345 accepts the write request at block 2250.

[0278] At block 2260, the head end server sends a read table request to the collector for the lost reports. The collector responds at block 2270 by fetching the appropriate reports the head end server has lost, optionally compressing them with suitable compression algorithms performed by a processor, and transmitting them to the head end server which receives the reports. At this point, the transfer and synchronization process is complete at block 2280.

[0279] The three-step synchronization process may also be useful for testing of the head end server or upgrading the head end server. Any number of head end servers may synchronize with the collector, and the last sequence number stored at each head end server may be different.

[0280] The words "herein," "above," "below," and words of similar import, when used in this application, shall refer to this application as a whole and not to any particular portions of this application. Where the context permits, words in the above Detailed Description using the singular or plural number may also include the plural or singular number respectively. The word "or," in reference to a list of two or more items, covers all of the following interpretations of the word: any of the items in the list, all of the items in the list, and any combination of the items in the list.

[0281] The above detailed description of embodiments of the invention is not intended to be exhaustive or to limit the invention to the precise form disclosed above. While specific embodiments of, and examples for, the invention are described above for illustrative purposes, various equivalent modifications are possible within the scope of the invention, as those skilled in the relevant art will recognize. Further any specific numbers noted herein are only examples: alternative implementations may employ differing values or ranges.

[0282] The teachings of the invention provided herein can be applied to other systems, not necessarily the system described above. The elements and acts of the various embodiments described above can be combined to provide further embodiments.

[0283] While the above description describes certain embodiments of the invention, and describes the best mode contemplated, no matter how detailed the above appears in text, the invention can be practiced in many ways. Details of the system may vary considerably in its implementation details, while still being encompassed by the invention disclosed herein. As noted above, particular terminology used

when describing certain features or aspects of the invention should not be taken to imply that the terminology is being redefined herein to be restricted to any specific characteristics, features, or aspects of the invention with which that terminology is associated. In general, the terms used in the following claims should not be construed to limit the invention to the specific embodiments disclosed in the specification, unless the above Detailed Description section explicitly defines such terms. Accordingly, the actual scope of the invention encompasses not only the disclosed embodiments, but also all equivalent ways of practicing or implementing the invention under the claims.

1. A system for communicating data from a node to a head end server, the system comprising:

a node for measuring, storing, and initiating transmission of data to a collector over a first network;

a collector for receiving the data over the first network and transmitting the data over a second network to the head end server;

the head end server being operative for receiving the data over the second network.

2. The system of claim **1**, further comprising the head end server.

3. The system of claim **1**, wherein the node includes a sensor for measuring utility usage and wherein utility is one of electricity, gas, and water.

4. The system of claim **1**, wherein the data is collected at a utility meter at a customer site.

5. The system of claim **1**, wherein the data is collected at a utility-regulating device at a customer site.

6. The system of claim **1**, wherein the data is collected at a utility-consuming device at a customer site.

7. The system of claim **1**, wherein the first network comprises a neighborhood area network.

8. The system of claim **1**, wherein the second network comprises a wide area network.

9. The system of claim **1**, wherein the collector is device agnostic.

10. The system of claim **1**, wherein the collector is data-structure agnostic.

11. The system of claim **1**, wherein the data is encrypted by the node.

12. The system of claim **11**, wherein the system head end server can decrypt the data but the collector can not decrypt the data.

13. The system of claim **1**, wherein the node further responds to a request for the data and transmits the data to the collector over the first network, and the collector further requests the data from the node over the first network.

14. The system of claim **1**, wherein the node includes at least one of a smart meter and a dumb meter.

15. The system of claim **1**, wherein the first and second networks support ANSI C12.22 protocols at an application layer level, and further wherein the application layer is supported by a transport layer.

16. A method of communicating data from a node to a head end server, the method comprising:

collecting, storing, and initiating transmission of data by the node to a collector over a first network;

receiving, by the collector, the data over the first network and transmitting, by the collector, the data over a second network to the head end server; and

receiving, by the head end server, the data over the second network.

17. The method of claim **16**, wherein the node further responds to a request for the data and transmits the data to the collector over the first network, and the collector further requests the data from the node and receives the data over the first network.

18. The method of claim **16**, the node includes a sensor for measuring utility usage and wherein utility is one of electricity, gas, and water.

19. The method of claim **16**, wherein the data is collected at one of the following: a utility meter at a customer site, a utility-regulating device at a customer site, and a utility-consuming device at a customer site.

20. The method of claim **16**, wherein the first network is a neighborhood area network.

21. The method of claim **16**, wherein the second network is a wide area network.

22. The method of claim **16**, wherein the collector is device-agnostic.

23. The method of claim **16**, wherein the collector is data-structure agnostic.

24. The method of claim **16**, wherein the data is encrypted by the node.

25. The method of claim **24**, wherein the system head end server can decrypt the data but the collector can not decrypt the data.

26. The method of claim **16**, wherein the node includes a smart meter or a dumb meter.

27. The method of claim **16**, wherein the first and second networks support ANSI C12.22 protocols at an application layer level, and further wherein the application layer is supported by a transport layer.

28. A collector device for operating in a system of the type that includes at least one node operative on a first network for storing data and a network server operative for receiving data on a second network, the collector device comprising:

a receiver unit for receiving the data from the node over the first network; and

a transmitter unit for transmitting data to the server over the second network;

the collector being node device and data-structure agnostic and operative to:

(i) request or pull data from a node device; or

(ii) accept information pushed from a node device when a node device initiates the transmission of data or a report to the collector; or

(iii) a combination of request or pull data from a node device and accept information pushed from a node.

29. A collector device as in claim **28**, wherein the node communicating data with the collector is a node device with the ability to communicate across a network.

30. A collector device as in claim **28**, wherein the node device has the ability to communicate across an AMI network and is further includes a utility-measuring capability.

31. A collector device as in claim **28**, wherein the utility measuring capability is provided by a utility measuring device selected from the set consisting of a utility meter, a utility consuming appliance, and a utility generating appliance.

32. A collector device as in claim **28**, wherein the node transmits data structures according to the specified protocol.

33. A collector device as in claim **28**, wherein the collector is adapted for receiving data from a plurality of different node device types, and the different node device types store and communicate data in different data structures.

34. A collector device as in claim **28**, wherein when the device-agnostic collector receives pushed data from a node device, and the device-agnostic collector does not need any special programs for communicating with different types of node devices about various types of data structures supported by the different node devices.

35. A collector device as in claim **28**, wherein:
the collector includes a plurality of individual ports for receiving data; and

the device-agnostic collector only requires the ability to receive data addressed and pushed to particular ones of its ports and to acknowledge the received data.

36. A collector device as in claim **35**, wherein the individual ports of the device-agnostic collector are reserved for receiving specific types of data.

37. A collector device as in claim **36**, wherein the different types of data includes different types of reports, and the different types of reports are selected from the set consisting of: batch-reporting type reports and real-time reporting reports.

38. A collector device as in claim **37**, wherein the collector further includes means for identifying the type of data received from a node device based on the collector port to which a node device addressed the data or report to the device-agnostic collector, the collector being adapted to process the received data or report and to transmit the report to the server appropriately based on the identification of data type received.

39. A collector device as in claim **28**, wherein the node is also operable to measure data.

40. A collector device as in claim **28**, wherein the node is further operable to initiate transmission of the data over the second network to the collector.

41. A collector device as in claim **28**, wherein the network server comprises a head end server and the head end server is operative to receive data from the collector over the second network.

42. A collector device as in claim **28**, wherein the first network includes at least a portion of the second network.

43. A collector device as in claim **28**, wherein the second network includes at least a portion of the first network.

44. A collector device as in claim **28**, wherein the first network and the second network are different networks.

45. A collector device as in claim **28**, wherein the node includes a sensor for measuring utility usage and wherein utility is one of electricity, gas, water, and information.

46. A collector device as in claim **28**, wherein the data is collected at a utility meter at a customer site.

47. A collector device as in claim **28**, wherein the data is collected at a utility-regulating device at a customer site.

48. A collector device as in claim **28**, wherein the data is collected at a utility-consuming device at a customer site.

49. A collector device as in claim **28**, wherein the first network comprises a neighborhood area network.

50. A collector device as in claim **28**, wherein the second network comprises a wide area network.

51. A collector device as in claim **28**, wherein the data is encrypted by the node.

52. A collector device as in claim **28**, wherein the system head end server can decrypt the data but the collector can not decrypt the data.

53. A collector device as in claim **28**, wherein the node further responds to a request for the data and transmits the

data to the collector over the first network, and the collector further requests the data from the node over the first network.

54. A collector device as in claim **28**, wherein the node includes at least one of a smart meter and a dumb meter.

55. A collector device as in claim **28**, wherein the first and second networks support ANSI C12.22 protocols at an application layer level, and further wherein the application layer is supported by a transport layer.

56. A method of operating a collector in a network environment including a plurality of different networked node devices having different device characteristics and different data structure characteristics and a networked head end server, the method comprising:

receiving a first data over the network environment from a first node device of a first node device type;

receiving a second data over the network environment from a second node device of a second node device type;

identifying the first data as the first data type and the second data as the second data type;

processing the received first data and second data according to whether it is the first data type or the second data type to generate a processed data; and

transmitting the processed data to the server over the network environment.

57. A method as in claim **56**, wherein the network environment includes a first network supporting communication between the collector and the first and second node device, and a second network supporting communication between the collector and the head end server.

58. A method as in claim **56**, wherein the collector is node device and node device data-structure agnostic and operative to:

(i) request or pull data from any node device; or

(ii) accept information pushed from any node device when a node device initiates the transmission of data or a report to the collector; or

(iii) a combination of request or pull data from a node device and accept information pushed from a node.

59. A method as in claim **58**, wherein the collector receives data from a plurality of different node device types, and the different node device types store and communicate data in different data structures.

60. A method as in claim **58**, wherein when the device-agnostic collector receives pushed data from a node device, the device-agnostic collector does not need any special programs for communicating with different types of node devices about various types of data structures supported by the different node devices.

61. A method as in claim **58**, wherein:

the collector includes a plurality of individual ports for receiving data; and

the collector receives data addressed and pushed to particular ones of its ports and acknowledges the received data.

62. A method as in claim **61**, wherein the individual ports of the device-agnostic collector are reserved for receiving specific types of data.

63. A method as in claim **62**, wherein the different types of data includes different types of reports, and the different types of reports are selected from the set consisting of: batch-reporting type reports and real-time reporting reports.

64. A method as in claim **63**, wherein the collector further identifies the type of data received from a node device based on the collector port to which a node device addressed the data or report to the device-agnostic collector, and the collec-

tor processes the received data or report and transmits the data or report to the server appropriately based on the identification of data type received.

65. The method of claim **57**, wherein the node further responds to a request for the data and transmits the data to the collector over the first network, and the collector further requests the data from the node and receives the data over the first network.

66. The method of claim **56**, the node includes a sensor for measuring utility usage and wherein utility is one of electricity, gas, and water.

67. The method of claim **56**, wherein the data is collected at one of the following: a node device having a utility meter at a customer site, a node device having a utility-regulating device at a customer site, and a node device having a utility-consuming device at a customer site.

68. The method of claim **57**, wherein the first network is a neighborhood area network.

69. The method of claim **57**, wherein the second network is a wide area network.

70. The method of claim **56**, wherein the data is encrypted by the node device.

71. The method of claim **70**, wherein the system head end server can decrypt the data but the collector can not decrypt the data.

72. The method of claim **56**, wherein the node includes a smart meter or a dumb meter.

73. The method of claim **57**, wherein the first and second networks support ANSI C12.22 protocols at an application layer level, and further wherein the application layer is supported by a transport layer.

74. A computer program stored in a computer readable form for execution within a processor and processor associated memory for controlling the operation of a collector for operating the collector in a network environment including a plurality of different networked node devices having different device characteristics and different data structure characteristics and a networked head end server, the operating or the controller implementing a method comprising:

- receiving a first data over the network environment from a first node device of a first node device type;
- receiving a second data over the network environment from a second node device of a second node device type;
- identifying the first data as the first data type and the second data as the second data type;
- processing the received first data and second data according to whether it is the first data type or the second data type to generate a processed data; and
- transmitting the processed data to the server over the network environment.

75. A system for communicating data from a node to a head end server, the system comprising:

- a node for measuring, storing, and initiating transmission of data to a collector over a first network; and
- a collector for receiving the data over the first network and transmitting the data over a second network to the head end server;
- the head end server being operative for receiving the data over the second network;
- the collector further comprising:
 - a receiver unit for receiving the data from the node over the first network; and
 - a transmitter unit for transmitting data to the server over the second network; and

the collector being node and data-structure agnostic and operative to:

- (i) request or pull data from a node, or
- (ii) accept information pushed from a node when a node initiates the transmission of data or a report to the collector, or
- (iii) a combination of request or pull data from a node and accept information pushed from a node.

76. A method of transmitting a report from a node to a head end server, the method comprising:

- pre-assigning one or more ports at a collector to receive particular types of reports;
- transmitting a data packet from the node to the collector over a first network, wherein the data packet includes a header containing a destination port of the collector and the report; and

transmitting the data packet from the collector to the head end server over a second network, wherein the collector follows reporting procedures based upon the destination port.

77. The method of claim **76**, wherein the particular types of reports include batch reports.

78. The method of claim **76**, wherein the particular types of reports include real-time reports.

79. The method of claim **76**, wherein the particular types of reports include on-demand reports.

80. The method of claim **76**, wherein the first network is a neighborhood area network that adheres to IEEE 802.15.4 protocol.

81. The method of claim **76**, wherein the second network is a wide area network that adheres to TCP/IP communications protocol.

82. The method of claim **76**, wherein the report adheres to ANSI C12 standards.

83. A method of transmitting data from a smart meter to a communications card for transmission to the collector, the method comprising:

- storing one or more lists of data structures in a register at the smart meter;
- storing a sequence number for each list in the register at the smart meter, wherein the sequence number is incremented by a predetermined number when a new piece of data is appended to the list;
- accessing and downloading new data from the one or more lists to a communications card, wherein the new data has corresponding sequence numbers that have not been previously downloaded;
- storing the data and the corresponding sequence number in memory;
- creating one or more reports from the new data at the communications card; and
- transmitting the one or more reports to the collector, wherein the communications card initiates the transmission.

84. The method of claim **83** further comprising downloading to the communications card other relevant information related to the data comprising data type and frequency of capture of the data.

85. The method of claim **83**, wherein the one or more reports adhere to ANSI C12 standards.

86. A method of using a reporting list at a node for transmission to a collector, the method comprising:

- creating one or more reports by the node;
- appending the one or more reports to a reporting list at the node;

transmitting to the collector a report on the reporting list immediately following a flagged report flagged by a last-transferred-pointer, wherein the transmission is initiated by the collector;
 receiving an acknowledgement from the collector; and
 updating the last-transferred-pointer;
 wherein the last-transferred-pointer is used to determine reports that have not been received by the collector.

87. The method of claim **86**, wherein the one or more reports adhere to ANSI C12 protocol.

88. The method of claim **86**, wherein the one or more reports are transmitted through a neighborhood area network that adheres to IEEE 802.15.4 protocol.

89. A system for transmitting a report from a node to a head end server, comprising:

means for pre-assigning one or more ports at a collector to receive particular types of reports;

means for transmitting a data packet from the node to the collector over a first network, wherein the data packet includes a header containing a destination port of the collector and the report; and

means for transmitting the data packet from the collector to the head end server over a second network, wherein the collector follows reporting procedures based upon the destination port.

90. The system of claim **89**, wherein the means for pre-assigning one or more ports at a collector to receive particular types of reports includes collector source code that has embedded within it collector port assignments.

91. The system of claim **89**, wherein the means for transmitting a data packet from the node to the collector over a first network includes a node radio for transmitting the data packet from the node to the collector over the first network.

92. The system of claim **89**, wherein the means for transmitting the data packet from the collector to the head end server over a second network includes a collector radio for transmitting the data packet from the collector to the head end server over the second network.

93. A system for transmitting a report from a node to a head end server, comprising:

collector computer code stored in a collector memory for pre-assigning one or more ports at a collector to receive particular types of reports;

a node transmitter for transmitting a data packet from the node to the collector over a first network, wherein the data packet includes a header containing a destination port of the collector and the report; and

a collector transmitter for transmitting the data packet from the collector to the head end server over a second network, wherein the collector follows reporting procedures based upon the destination port.

94. A system for transmitting data from a smart meter to a communications card for transmission to the collector, comprising:

means for storing one or more lists of data structures at the smart meter;

means for storing a sequence number for each list at the smart meter, wherein the sequence number is incremented by a predetermined number when a new piece of data is appended to the list;

means for accessing and downloading new data from the one or more lists to a communications card, wherein the new data has corresponding sequence numbers that have not been previously downloaded;

means for storing the data and the corresponding sequence number;

means for creating one or more reports from the new data at the communications card; and

means for transmitting the one or more reports to the collector, wherein the communications card initiates the transmission.

95. The system of claim **94**, wherein the means for storing one or more lists of data structures at the smart meter includes a register.

96. The system of claim **94**, wherein the means for storing a sequence number for each list at the smart meter includes a register.

97. The system of claim **94**, wherein the means for means for accessing and downloading new data from the one or more lists to a communications card includes a processor for accessing new data and a receiver and for downloading the new data to a memory.

98. The system of claim **94**, wherein the means for storing the data and the corresponding sequence number includes a memory.

99. The system of claim **94**, wherein the means for creating one or more reports from the new data at the communications card includes a processor for creating one or more reports.

100. The system of claim **94**, wherein the means for transmitting the one or more reports to the collector includes a node radio for transmitting the one or more reports to the collector.

101. A system for transmitting data from a smart meter to a communications card for transmission to the collector, comprising:

a register for storing one or more lists of data structures at the smart meter and a sequence number for each list at the smart meter, wherein the sequence number is incremented by a predetermined number when a new data structure is appended to the list;

a processor for accessing data structures on the list and creating one or more reports from the new data at the communications card;

a receiver and a memory for downloading the new data, wherein the new data has corresponding sequence numbers that have not been previously downloaded, and further wherein the memory stores the new data and the corresponding sequence number; and

a node transmitter for transmitting the one or more reports to the collector, wherein the communications card initiates the transmission.

102. A system for using a reporting list at a node for transmission to a collector comprising:

means for creating one or more reports by the node;

means for appending the one or more reports to a reporting list at the node;

means for transmitting to the collector a report on the reporting list immediately following a flagged report flagged by a last-transferred-pointer;

means for receiving an acknowledgement from the collector; and

means for updating the last-transferred-pointer;

wherein the last-transferred-pointer is used to determine reports that have not been received by the collector.

103. The system of claim **102**, wherein the means for creating one or more reports by the node includes a node processor for generating one or more reports.

104. The system of claim **102**, wherein the means for appending the one or more reports to a reporting list at the node includes a node processor for appending the one or more reports to a reporting list stored in a memory.

105. The system of claim **102**, wherein the means for transmitting to the collector a report on the reporting list immediately following a flagged report flagged by a last-transferred-pointer includes a node radio for transmitting to the collector a report on the reporting list immediately following a flagged report flagged by a last-transferred-pointer.

106. The system of claim **102**, wherein the means for receiving an acknowledgement from the collector includes a node radio for receiving an acknowledgement from the collector.

107. The system of claim **102**, wherein the means for updating the last-transferred-pointer includes a node processor for updating the last-transferred-pointer stored in memory.

108. A system for using a reporting list at a node for transmission to a collector comprising:

a node processor for creating one or more reports by the node, appending the one or more reports to a reporting list at the node, and updating a last-transferred-pointer; a node transmitter for transmitting to the collector a report on the reporting list immediately following a flagged report flagged by the last-transferred-pointer; and a node receiver for receiving an acknowledgement from the collector;

wherein the last-transferred-pointer is used to determine reports that have not been received by the collector.

109. A computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of transmitting a report from a node to a head end server, the method comprising:

pre-assigning one or more ports at a collector to receive particular types of reports;

transmitting a data packet from the node to the collector over a first network, wherein the data packet includes a header containing a destination port of the collector and the report; and

transmitting the data packet from the collector to the head end server over a second network, wherein the collector follows reporting procedures based upon the destination port.

110. A computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of transmitting data from a smart meter to a communications card for transmission to the collector, the method comprising:

storing one or more lists of data structures in a register at the smart meter;

storing a sequence number for each list in the register at the smart meter, wherein the sequence number is incremented by a predetermined number when a new piece of data is appended to the list;

accessing and downloading new data from the one or more lists to a communications card, wherein the new data has corresponding sequence numbers that have not been previously downloaded;

storing the data and the corresponding sequence number in memory;

creating one or more reports from the new data at the communications card; and

transmitting the one or more reports to the collector, wherein the communications card initiates the transmission.

111. A computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of using a reporting list at a node for transmission to a collector, the method comprising:

creating one or more reports by the node;

appending the one or more reports to a reporting list at the node;

transmitting to the collector a report on the reporting list immediately following a flagged report flagged by a last-transferred-pointer, wherein the transmission is initiated by the collector;

receiving an acknowledgement from the collector; and

updating the last-transferred-pointer;

wherein the last-transferred-pointer is used to determine reports that have not been received by the collector.

112. A node device for initiating transmission of data to a collector comprising:

a data input/output module for collecting the data;

a memory for storing the data;

a processor for creating a report from the data; and

a radio for transmitting the report to the collector;

wherein the transmission is initiated by the node.

113. The node device of claim **112**, wherein the input/output module is a sensor.

114. The node device of claim **112**, wherein the input/output module is a thermostat.

115. The node device of claim **112**, wherein the radio is a communications card.

116. The node device of claim **112**, wherein the report adheres to ANSI C12 standards.

117. The node device of claim **112**, further comprising:

a primary power supply for supplying power to the input/output module, the memory, the processor, and the radio; and

a secondary power supply for supplying power to the input/output module, the memory, the processor, and the radio if the primary power supply is interrupted.

118. A collector for receiving a data packet over a first network initiated from a node comprising:

a first radio for receiving the data packet initiated by the node over the first network;

a second radio for transmitting the data packet over a second network to a head end server; and

a memory for storing the data packet to be transmitted to the head end server and used by the processor.

119. The collector of claim **118**, wherein the data packet is a report adhering to ANSI C12 standards.

120. The collector of claim **118**, wherein the first network is a neighborhood area network adhering to IEEE 802.15.4 protocol.

121. The collector of claim **118**, wherein the second network is a wide area network adhering to TCP/IP communications protocol.

122. A method of operating a node that initiates a report to a collector comprising:

collecting data through an input device;

storing the data in a memory;

creating the report from the data; and

transmitting the report to the collector over a first network; wherein the report is initiated by the node.

123. The method of claim **122**, wherein the report adheres to ANSI C12 standards.

124. The method of claim **122**, wherein the first network is a neighborhood area network adhering to IEEE 802.15.4 protocol.

125. A method of operating a collector that receives a data packet initiated by a node comprising:

- receiving the data packet initiated by the node over a first network addressed to a particular collector port, wherein the data packet comprises a report;
- processing the report according to pre-assigned procedures corresponding to the particular collector port; and
- transmitting the report to a head end over a second network.

126. The method of claim **125**, wherein the first network is a neighborhood area network adhering to IEEE 802.15.4 protocol.

127. The method of claim **125**, wherein the second network is a wide area network adhering to TCP/IP communications protocol.

128. The method of claim **125**, wherein the report adheres to ANSI C12 standards.

129. The method of claim **125**, wherein the collector is device-agnostic and data structure-agnostic.

130. A computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of operating a node that initiates a report to a collector comprising:

- collecting data through an input device;
- storing the data in a memory;
- creating the report from the data; and
- transmitting the report to the collector over a first network; wherein the report is initiated by the node.

131. A computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of operating a collector that receives a data packet initiated by a node comprising:

- receiving the data packet initiated by the node over a first network addressed to a particular collector port, wherein the data packet comprises a report;
- processing the report according to pre-assigned procedures corresponding to the particular collector port; and
- transmitting the report to a head end over a second network.

132. A method of communicating between a plurality of nodes and a collector using checkpoints, comprising:

- generating a substantially random time slot within a pre-scheduled reporting time window for each node; and
- transmitting data packets from each node to the collector at the node's substantially random time slot within the prescheduled reporting time window.

133. The method of claim **132** further comprising acknowledging receipt of each data packet by the collector.

134. The method of claim **132**, wherein the substantially random time slot generated for a particular node is dependent upon an address of the node.

135. The method of claim **132**, wherein the substantially random time slot generated for a particular node is dependent upon a manufacturer's serial number for a device at the node, wherein the device is selected from a group consisting of: a meter, a utility-consumption device, and a utility-measuring device.

136. The method of claim **135**, wherein the substantially random time slot generated for a particular node is dependent upon a device reading at the particular node.

137. A method of transmitting data packets from a plurality of nodes to a collector, comprising:

- generating a first substantially random time slot within a prescheduled reporting time window for each node;
- transmitting data packets from each node to the collector at the node's substantially random time slot within the prescheduled reporting time window;
- transmitting an acknowledgement from the collector for each received data packet;
- generating a second substantially random time slot within a second prescheduled reporting time window for each node that does not receive an acknowledgement from the collector; and
- re-transmitting data packets from each node that did not receive the acknowledgement from the collector at the node's second substantially random time slot within the second prescheduled reporting time window.

138. The method of claim **137**, wherein the first substantially random time slot and the second substantially random time slot generated for a particular node is based on an address of the node.

139. The method of claim **137**, wherein the first substantially random time slot and the second substantially random time slot generated for a particular node is based on a manufacturer's serial number for a device at the node, wherein the device is selected from a group consisting of: a meter, a utility-consumption device, a utility-measuring device, and combinations of these.

140. The method of claim **139**, wherein the first substantially random time slot and the second substantially random time slot generated for a particular node is based on a device reading at the particular node.

141. A system for communicating between a plurality of nodes and a collector using checkpoints, comprising:

- means for generating a substantially random time slot within a pre-scheduled reporting time window for each node; and
- means for transmitting data packets from each node to the collector at the node's substantially random time slot within the prescheduled reporting time window.

142. The system of claim **141**, further comprising means for acknowledging receipt of each data packet by the collector.

143. The system of claim **141**, wherein the substantially random time slot generated for a particular node is dependent upon an address of the node.

144. The system of claim **141**, wherein the substantially random time slot generated for a particular node is dependent upon a manufacturer's serial number for a node device at the node, wherein the node device is selected from a group consisting of: a meter, a utility-consumption device, a utility-measuring device, and combinations of these.

145. The system of claim **141**, wherein the substantially random time slot generated for a particular node is dependent upon a device reading at the particular node.

146. The system of claim **141**, wherein the means for generating a substantially random time slot within a pre-scheduled reporting time window for each node comprises a random time slot number generator.

147. The system of claim **141** wherein the means for transmitting data packets from each node to the collector at the node's substantially random time slot comprises a node radio for transmitting data packets from each node to the collector at the node's substantially random time slot.

148. The system of claim **141** wherein the means for acknowledging receipt of each data packet by the collector comprises a collector radio for acknowledging receipt of each data packet by the collector.

149. A system for communicating between a plurality of nodes and a collector using checkpoints, comprising:

a substantially random time slot generator for generating a substantially random time slot within a pre-scheduled reporting time window for each node; and

a node transmitter for transmitting data packets from each node to the collector at the node's substantially random time slot within the prescheduled reporting time window.

150. The system of claim **149**, further comprising a collector transmitter for acknowledging receipt of each data packet by the collector.

151. The system of claim **149**, wherein the substantially random time slot generated for a particular node is dependent upon an address of the node.

152. The system of claim **149**, wherein the substantially random time slot generated for a particular node is dependent upon a manufacturer's serial number for a node device at the node, wherein the node device is selected from a group consisting of: a meter, a utility-consumption device, a utility-measuring device, and combinations of these.

153. The system of claim **149**, wherein the substantially random time slot generated for a particular node is dependent upon a device reading at the particular node.

154. A system for transmitting data packets from a plurality of nodes to one collector comprising:

means for generating a first substantially random time slot within a prescheduled reporting time window for each node;

means for transmitting data packets from each node to the collector at the node's substantially random time slot within the prescheduled reporting time window;

means for generating a second substantially random time slot within a second prescheduled reporting time window at each node that does not receive an acknowledgement from the collector; and

means for re-transmitting data packets from each node that did not receive the acknowledgement from the collector at the node's second substantially random time slot within the second prescheduled reporting time window.

155. The system as in claim **154** wherein the means for generating a first substantially random time slot and a second substantially random time slot within a prescheduled reporting time window for each node includes a random time slot number generator.

156. The system as in claim **154**, wherein the means for transmitting data packets from each node to the collector at the node's first substantially random time slot and second substantially random time slot includes a transmitter for transmitting data packets from each node to the collector at the node's random time slot.

157. The system as in claim **154**, wherein the means for re-transmitting data packets from each node that did not receive the acknowledgement from the collector at the node's second substantially random time slot includes a transmitter for re-transmitting data packets from each node that did not receive the acknowledgement from the collector at the node's substantially second random time slot.

158. A system for transmitting data packets from a plurality of nodes to one collector comprising:

a substantially random time slot generator for generating a first substantially random time slot within a prescheduled reporting time window for each node and for generating a second substantially random time slot within a second prescheduled reporting time window at each node that does not receive an acknowledgement from the collector; and

a transmitter for transmitting data packets from each node to the collector at the node's substantially random time slot within the prescheduled reporting time window and for re-transmitting data packets from each node that did not receive the acknowledgement from the collector at the node's second substantially random time slot within the second prescheduled reporting time window.

159. A computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of communicating between a plurality of nodes and a collector using checkpoints, comprising:

generating a substantially random time slot within a prescheduled reporting time window for each node; and transmitting data packets from each node to the collector at the node's substantially random time slot within the prescheduled reporting time window.

160. A computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of transmitting data packets from a plurality of nodes to a collector, comprising:

generating a first substantially random time slot within a prescheduled reporting time window for each node;

transmitting data packets from each node to the collector at the node's substantially random time slot within the prescheduled reporting time window;

transmitting an acknowledgement from the collector for each received data packet;

generating a second substantially random time slot within a second prescheduled reporting time window for each node that does not receive an acknowledgement from the collector; and

re-transmitting data packets from each node that did not receive the acknowledgement from the collector at the node's second substantially random time slot within the second prescheduled reporting time window.

161. A method of synchronizing a data request between a head end server and a collector comprising:

transmitting from the head end server a first read request for a first table from the collector, wherein elements in the first table correspond to nodes serviced by the collector;

transmitting the first table from the collector to the head end server;

transmitting from the head end server a write request, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last received report from each node stored in a database at the head end server;

accepting the write request if the write request is proper;

transmitting from the head end server a second read request for new reports; and

transmitting the new reports from the collector to the head end server.

162. The method of claim **161**, wherein the elements of the first table comprise ApTitles corresponding to each node.

163. The method of claim **162**, further comprising compressing the first table prior to transmitting to the head end server.

164. The method of claim **162**, further comprising compressing new reports prior to transmitting to the head end server.

165. The method of claim **162**, further comprising authenticating the head end server.

166. The method of claim **162**, wherein the write request is proper if the head end server has permission to access the new reports.

167. A system for synchronizing a data request between a head end server and a collector comprising:

means for transmitting from the head end server a first read request for a first table from the collector, wherein elements in the first table correspond to nodes serviced by the collector;

means for transmitting the first table from the collector to the head end server;

means for transmitting from the head end server a write request, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last received report from each node stored in a database at the head end server;

means for accepting the write request if the write request is proper;

means for transmitting from the head end server a second read request for new reports; and

means for transmitting the new reports from the collector to the head end server.

168. The system of claim **167**, wherein the elements of the first table comprise ApTitles corresponding to each node.

169. The system of claim **167**, further comprising means for compressing the first table prior to transmitting to the head end server.

170. The system of claim **167**, further comprising means for compressing new reports prior to transmitting to the head end server.

171. The system of claim **167**, further comprising means for authenticating the head end server.

172. The system of claim **167**, wherein the write request is proper if the head end server has permission to access the new reports.

173. The system of claim **167**, wherein the means for transmitting a first read request from the head end server for a first table from the collector, wherein elements in the first table correspond to nodes serviced by the collector includes a head end server transmitter for transmitting the first read request.

174. The system of claim **167**, wherein the means for transmitting the first table from the collector to the head end server includes a head end server transmitter for transmitting the first table to the head end server.

175. The system of claim **167**, wherein the means for transmitting a write request from the head end server includes a head end server transmitter for transmitting the write request.

176. The system of claim **167**, wherein the means for accepting the write request if the write request is proper includes a collector processor for accepting the write request.

177. The system of claim **167**, wherein the means for transmitting from the head end server a second read request for new reports includes a head end server transmitter for transmitting a second read request for new reports.

178. The system of claim **167**, wherein the means for transmitting the new reports from the collector to the head end server includes a collector transmitter for transmitting new reports.

179. The system of claim **167**, wherein the means for compressing first table prior to transmitting to the head end server includes a processor for applying compression algorithms for compressing the first table.

180. The system of claim **167**, wherein the means for compressing new reports prior to transmitting to the head end server includes a processor for executing compression algorithms for compressing new reports.

181. The system of claim **167**, wherein the means for authenticating the head end server includes a processor for authenticating the head end server.

182. A system for synchronizing a data request between a head end server and a collector comprising:

(a) a head end server transmitter for transmitting:

a first read request for a first table from the collector, wherein elements in the first table correspond to nodes serviced by the collector;

a write request, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last received report from each node stored in a database at the head end server; and

a second read request for new reports;

(b) a collector transmitter for transmitting:

the first table to the head end server; and

the new reports to the head end server; and

(c) a collector receiver and a collector processor for accepting the write request if the write request is proper.

183. The system of claim **182**, wherein the elements of the first table comprise ApTitles corresponding to each node.

184. The system of claim **182**, wherein the collector processor compresses the first table using one or more compression algorithms prior to transmitting to the head end server.

185. The system of claim **182**, wherein the collector processor compresses new reports using one or more compression algorithms prior to transmitting to the head end server.

186. The system of claim **182**, wherein the collector processor authenticates the head end server based upon information stored in a collector memory.

187. The system of claim **182**, wherein the write request is proper if the head end server has permission to access the new reports.

188. A computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of synchronizing a data request between a head end server and a collector comprising:

transmitting from the head end server a first read request for a first table from the collector, wherein elements in the first table correspond to nodes serviced by the collector;

transmitting the first table from the collector to the head end server;

transmitting from the head end server a write request, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last received report from each node stored in a database at the head end server;

accepting the write request if the write request is proper;

transmitting from the head end server a second read request for new reports; and

transmitting the new reports from the collector to the head end server.

189. A method of synchronizing a data request between a head end server and a collector comprising:

transmitting from the head end server a first read request for a first table from the collector, wherein elements in the first table correspond to nodes serviced by the collector;

receiving by the head end server the first table from the collector;

transmitting from the head end server a write request, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last received report from each node stored in a database at the head end server;

transmitting from the head end server a second read request for new reports; and

receiving the new reports from the collector.

190. A method of synchronizing a data request between a head end server and a collector comprising:

receiving by the collector from the head end server a first read request for a first table, wherein elements in the first table correspond to nodes serviced by the collector;

transmitting the first table from the collector to the head end server;

receiving by the collector from the head end server a write request, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last received report from each node stored in a database at the head end server;

accepting by the collector the write request if the write request is proper;

receiving by the collector from the head end server a second read request for new reports; and

transmitting the new reports from the collector to the head end server.

191. A computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of synchronizing a data request between a head end server and a collector comprising:

transmitting from the head end server a first read request for a first table from the collector, wherein elements in the first table correspond to nodes serviced by the collector;

receiving by the head end server the first table from the collector;

transmitting from the head end server a write request, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last received report from each node stored in a database at the head end server;

transmitting from the head end server a second read request for new reports; and

receiving the new reports from the collector.

192. A computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of synchronizing a data request between a head end server and a collector comprising:

receiving by the collector from the head end server a first read request for a first table, wherein elements in the first table correspond to nodes serviced by the collector;

transmitting the first table from the collector to the head end server;

receiving by the collector from the head end server a write request, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last received report from each node stored in a database at the head end server;

accepting by the collector the write request if the write request is proper;

receiving by the collector from the head end server a second read request for new reports; and

transmitting the new reports from the collector to the head end server.

193. The method of claim **162**, wherein the ApTitle includes the application process title of a node and corresponds to the address of the node; and wherein the ApTitle of a node within a particular collector sub-network uniquely identifies that node.

194. The system of claim **168**, wherein the ApTitle includes the application process title of a node and corresponds to the address of the node; and wherein the ApTitle of a node within a particular collector sub-network uniquely identifies that node.

195. The computer program of claim **183**, wherein the ApTitle includes the application process title of a node and corresponds to the address of the node; and wherein the ApTitle of a node within a particular collector sub-network uniquely identifies that node.

196. A method of recovering loss of data at a head end server's database comprising:

transmitting a first read request from the head end server for a first table from the collector, wherein elements in the first table correspond to nodes serviced by the collector;

transmitting the first table from the collector to the head end server; transmitting a write request from the head end server, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last available report from each node stored in a database at the head end server after the loss of data;

accepting the write request if the write request is proper;

transmitting from the head end server a second read request for lost reports; and

transmitting the lost reports from the collector to the head end server.

197. The method of claim **196**, wherein the elements of the first table comprise ApTitles corresponding to each node.

198. The method of claim **196**, further comprising compressing the first table prior to transmitting to the head end server.

199. The method of claim **196**, further comprising compressing lost reports prior to transmitting to the head end server.

200. The method of claim **196**, further comprising authenticating the head end server.

201. The method of claim **197**, wherein the write request is proper if the head end server has permission to access the lost reports.

202. A method of recovering from a collector failure comprising:

seeking a new collector by a node associated with the failed collector; and

transmitting one or more reports from the node's reporting list.

203. The method of claim **202**, wherein transmitting reports from the node's reporting list includes reports immediately following a flagged report flagged by a last-transferred-pointer.

204. The method of claim **202**, further comprising:
receiving an acknowledgment from the new collector for each report received by the new collector; and
updating the last-transferred-pointer to flag the last report acknowledged by the new collector.

205. A system for recovering from a collector failure comprising:

means for seeking a new collector by a node associated with the failed collector; and

means for transmitting one or more reports from the node's reporting list.

206. The system of claim **205**, wherein the means for seeking the new collector by the node includes a receiver for detecting active networks and receiving responses from the new collector, a transmitter for transmitting a request for association, and a processor to run software for associating with a new collector.

207. The system of claim **205**, wherein the means for transmitting reports from the node's reporting list includes a transmitter for transmitting reports.

208. The system of claim **205**, wherein transmitting reports from the node's reporting list includes reports immediately following a flagged report flagged by a last-transferred-pointer.

209. The system of claim **205**, further comprising:

means for receiving an acknowledgment from the new collector for each report received by the new collector; and

means for updating the last-transferred-pointer to flag the last report acknowledged by the new collector.

210. The system of claim **209**, wherein the means for acknowledging receipt of the transmitted reports by the new collector includes a receiver for acknowledging receipt of the transmitted reports by the new collector.

211. The system of claim **209**, wherein the means for updating the last-transferred-pointer to flag the last report acknowledged by the new collector includes a processor for updating the last-transferred-pointer.

212. A system for recovering from a collector failure comprising:

a radio and a microcontroller for seeking a new collector by a node associated with the failed collector and transmitting one or more reports from the node's reporting list.

213. The system of claim **212**, further comprising:

a processor for updating the last-transferred-pointer to flag the last report acknowledged by the new collector; and

wherein the radio receives an acknowledgment from the new collector for each report received by the new collector.

214. The system of claim **212** wherein transmitting reports from the node's reporting list includes reports immediately following a flagged report flagged by a last-transferred-pointer.

215. A system for recovering loss of data at a head end server comprising:

means for transmitting a first read request from the head end server for a first table from the collector, wherein elements in the first table correspond to nodes serviced by the collector;

means for transmitting the first table from the collector to the head end server;

means for transmitting a write request from the head end server, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last available report from each node stored in a database at the head end server after the loss of data;

means for accepting the write request if the write request is proper;

means for transmitting from the head end server a second read request for lost reports; and

means for transmitting the lost reports from the collector to the head end server.

216. The system of claim **215**, wherein the elements of the first table comprise ApTitles corresponding to each node.

217. The system of claim **215**, wherein the means for transmitting a first read request from the head end server for a first table from the collector, wherein elements in the first table correspond to nodes serviced by the collector includes a head end server transmitter for transmitting a first request.

218. The system of claim **215**, wherein the means for transmitting the first table from the collector to the head end server includes a collector transmitter for transmitting the first table.

219. The system of claim **215**, wherein the means for transmitting a write request from the head end server, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last available report from each node stored in a database at the head end server after the loss of data includes a head end server transmitter for transmitting the write request.

220. The system of claim **215**, wherein the means for accepting the write request if the write request is proper includes a collector processor for accepting the write request.

221. The system of claim **215**, wherein the means for transmitting from the head end server a second read request for lost reports includes a head end server transmitter for transmitting the second read request.

222. The system of claim **215**, wherein the means for transmitting the lost reports from the collector to the head end server includes a collector transmitter for transmitting the lost reports.

223. A system for recovering loss of data at a head end server comprising:

a head end server transmitter for transmitting:

a first read request for a first table from the collector, wherein elements in the first table correspond to nodes serviced by the collector;

a write request, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last available report from each node stored in a database at the head end server after the loss of data; and a second read request for lost reports; and
a collector transmitter for transmitting:
the first table from the collector to the head end server;
and
the lost reports from the collector to the head end server;
and
a collector receiver and a collector processor for accepting the write request if the write request is proper.

224. The system of claim **223**, wherein the elements of the first table comprise ApTitles corresponding to each node.

225. A computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of recovering loss of data at a head end server's database comprising:
transmitting a first read request from the head end server for a first table from the collector, wherein elements in the first table correspond to nodes serviced by the collector;

transmitting the first table from the collector to the head end server;
transmitting a write request from the head end server, wherein the write request includes a second table comprising last sequence numbers, and each last sequence number uniquely corresponds to a last available report from each node stored in a database at the head end server after the loss of data;
accepting the write request if the write request is proper;
transmitting from the head end server a second read request for lost reports; and
transmitting the lost reports from the collector to the head end server.

226. A computer program product stored in a computer readable media for execution in a processor and memory coupled to the processor for performing a method of recovering from a collector failure comprising:
seeking a new collector by a node associated with the failed collector; and
transmitting one or more reports from the node's reporting list.

* * * * *