(12)Indian Patent Application

---

(21) Application Number: 202141036738

(22) Filing Date: 13/08/2021 (43) Publication Date: 17/02/2023

(71) Applicant(s): L&T TECHNOLOGY SERVICES LIMITED

(72) Inventor(s): Kumbhalkar, Kaustubh
De, Nilanjan
Srinivas, Pavan Kumar

(51) International Classifications: H04L 29/08 H04L 29/06 H04W 88/16 H04L 12/66 H04W 4/70

(54) Title: OUT OF BAND (OOB) SYSTEM AND METHOD FOR MANAGING INTERNET OF THINGS (IoT) GATEWAYS

(57) Abstract: OOB method for operation management of a set of IoT gateways is provided. The OOB method includes registering at least one IoT gateway with IoT hub and an OOB IoT registration service. A central gateway health monitor is also registered with OOB IoT registration service to obtain OOB client list associated with set of IoT gateways. The OOB method includes connecting at least one IoT gateway with central gateway health monitor configured to have OOB management of set of IoT gateways using separate communication channel for centrally located logs and operating monitoring system for the set of IoT gateways. The OOB method includes obtaining OOB client list, connecting at least one IoT gateway with one or more of set of IoT gateways, via local communication network and transmitting data from at least one IoT gateway to central gateway health monitor for operation management of set of IoT gateways at edge.

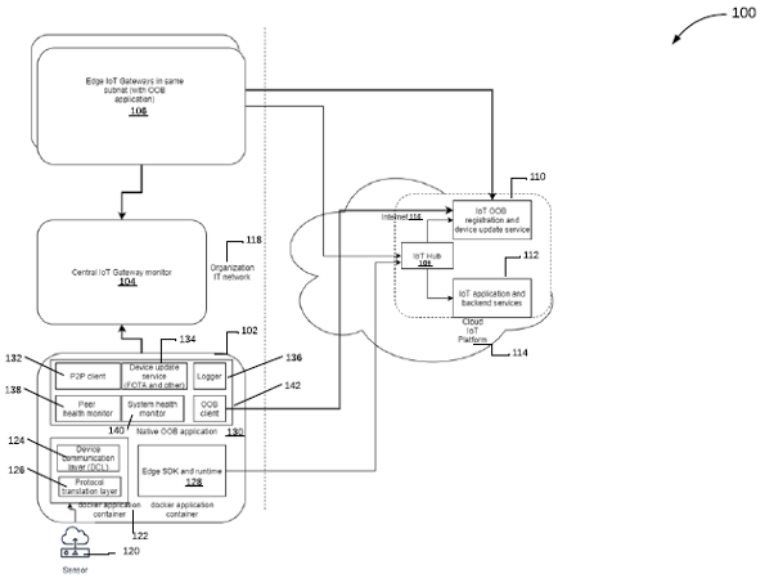

Fig 1

FORM 2

THE PATENTS ACT 1970
(39 OF 1970)
&
The Patent Rules, 2003

Complete Specification

(See Section 10 and Rule 13)

TITLE OF THE INVENTION

OUT OF BAND (OOB) SYSTEM AND METHOD FOR MANAGING INTERNET OF THINGS (IoT) GATEWAYS

APPLICANT(S)

(a) NAME : **L&T TECHNOLOGY SERVICES LIMITED**

(b) NATIONALITY : **INDIAN**

(c) ADDRESS : DLF IT SEZ Park, 2nd Floor – Block 3

1/124, Mount Poonamallee Road,

Ramapuram, Chennai – 600 089,

INDIA.

3. PREAMBLE TO THE DESCRIPTION

COMPLETE

The following specification describes the invention and the manner in which it is to be performed

DESCRIPTION

Technical Field

[001] This disclosure relates generally to Internet of Things (IoT) management, and more particularly relates to Out of Band (OOB) system and method for cloud agnostic configuration, monitoring and maintenance of IoT gateways.

Background

[001] Typically, IoT devices are deployed in a smart city or in a smart industrial IoT solution. IoT devices may correspond to sensors and IoT gateways that are deployed on a site (or edge) to collect machine data and environmental data. Generally, the IoT devices may have insufficient memory to store the machine data and the environmental data.

[002] An IoT gateway may correspond to a computer machine deployed at edge near sensors or directly connected to the sensors for data collection and uses multiple protocols. The IoT gateway may be without a graphical user interface, however, may require applications to be run for data collection from peripherals and for transmitting data to a remote server for analysis. The IoT gateway (also referred as an edge device) are deployed remotely and are connected to IoT applications deployed in a remote cloud. The IoT gateway may be prone to software and hardware failures in their lifetime of working and are also targets for physical and software hacking. The IoT gateway and IoT device provisioning may be used to deploy modules at edge from the cloud, however, this does not provide on-premises control over the monitoring and logging required by a local organization IT policy. In certain scenarios, sidecar and service mesh paradigm may be used in microservices management of the cloud. However, they are specific to technologies in cloud and not for embedded edge systems.

[003] In certain other scenarios, IoT systems have issues with internet connectivity, bandwidth and reliability of communication for logging and monitoring in the cloud. Though IoT platforms provide distributed services and application architecture in the cloud, services on the edge are monolithic. Further, industrial IT networks may be connected to Internet and the cloud for providing communication channel to the cloud, however, it is highly constrained due to IT firewall policy, network segregation (Internet vs Intranet and OT network vs IT

network). Therefore, sensitive secure data, like logs, detailed sensor configuration cannot be shared on cloud deployed IoT platforms.

[004] Accordingly, there is a need for a method and system for cloud agnostic configuration, monitoring and maintenance of the IoT gateways in a secure manner.

SUMMARY

[005] An Out of Band (OOB) method for operation management of a set of Internet of Things (IoT) gateways deployed at edge is disclosed. The OOB method includes registering at least one IoT gateway from the set of IoT gateways with an OOB IoT registration service. A central gateway health monitor is also registered with the OOB IoT registration service to obtain OOB client list associated with the set of IoT gateways. The OOB IoT registration service manages gateway-to-gateway communication, via a local communication network. The OOB method further includes connecting the at least one IoT gateway from the set of IoT gateways with the central gateway health monitor after registration. In accordance with an embodiment, the central gateway health monitor is configured to have OOB management of the set of IoT gateways using a separate communication channel for centrally located logs associated with at least one of the set of IoT gateways and an operating monitoring system for the set of IoT gateways. The OOB method further includes obtaining an OOB client list at the at least one IoT gateway from the central gateway health monitor, wherein the OOB client list corresponds to a list of IoT gateways registered with the OOB IoT registration service. The OOB method further includes connecting the at least one IoT gateway with one or more of the set of IoT gateways, via the local communication network for real time log data and system health data associated with the one or more of the set of IoT gateways, based on the OOB client list. The OOB method further includes transmitting the real time log data and system health data associated with the one or more of the set of IoT gateways from the at least one IoT gateway to the central gateway health monitor for the operation management of the set of IoT gateways at the edge.

[006] It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the invention, as claimed.

BRIEF DESCRIPTION OF THE DRAWINGS

[007] The accompanying drawings, which are incorporated in and constitute a part of this disclosure, illustrate exemplary embodiments and, together with the description, serve to explain the disclosed principles.

[008] FIG. 1 is a block diagram that illustrates an overall topology when deploying edge IoT gateways and architecture of the edge IoT gateway software implementation, in accordance with an embodiment of the disclosure.

[009] FIG. 2 is a block diagram that illustrates an exemplary scenario of Out of Band (OOB) method for Peer to Peer (P2P) communication for collecting and aggregating data amongst edge IoT gateways, in accordance with an embodiment of the disclosure.

[010] FIG. 3A-3B collectively illustrate an exemplary registration call flow for IoT Gateway and Central IoT gateway monitor within premises of organization IT network, with the IoT platform in cloud, in accordance with an embodiment of the disclosure.

DETAILED DESCRIPTION

[011] Exemplary embodiments are described with reference to the accompanying drawings. Wherever convenient, the same reference numbers are used throughout the drawings to refer to the same or like parts. While examples and features of disclosed principles are described herein, modifications, adaptations, and other implementations are possible without departing from the spirit and scope of the disclosed embodiments. It is intended that the following detailed description be considered as exemplary only, with the true scope and spirit being indicated by the following claims. Additional illustrative embodiments are listed below.

[012] The following described implementations may be found in the disclosed system and OOB method for cloud agnostic OOB configuration, maintenance and monitoring of IoT gateways. Exemplary aspects of the disclosure provide a robust and fail-safe, redundant OOB method for operation management of the IoT gateways deployed at edge, improved redundancy and reliability of data at edge. The disclosed system and OOB method implements a local communication sidecar module to deploy components of an application into a separate process or container to provide isolation and encapsulation. The sidecar module may support processes or services that are deployed with a primary application. Applications and services often

require related functionality, such as monitoring, logging, configuration, and networking services. These peripheral tasks can be implemented as separate components or services by using the local communication sidecar module.

[013] The disclosed system and OOB method may have ability to implement rapid decision and control with gateway-to-gateway communication data aggregation at edge. The disclosed system and OOB method may have advantage in monitoring sensors and other gateways in same location and network. Exemplary aspects of the disclosure provide may provide peer to peer (P2P) communication method for local data aggregation for rapid decision and control, OOB monitoring method to deliver device update management with any cloud IoT platform, configurable multi-protocol support as an IoT edge module. The disclosed OOB method improves fault tolerance for the IoT gateway with the help of local health monitor. Exemplary aspects of the disclosure provide a deployment architecture for an industrial customer that may be in-line with their Information Technology (IT) and Operational Technology (OT) network policies and IoT monitoring can be done within an organization IT network boundary without sharing security sensitive data.

[014] The disclosed system may facilitate operational monitoring of the IoT devices in the edge network with benefits of fast response time, low internet and cloud usage costs, secure IT management because of no exposing of gateway details of software and hardware configuration, triangulation of failure through integration with peer-to-peer monitoring and use of network management tools.

[015] FIG. 1 is a block diagram that illustrates an overall topology of gateways deployment and gateway software architecture implementation, in accordance with an embodiment of the disclosure.

[016] With reference to FIG.1, there is shown a block diagram 100 that includes an edge IoT gateway 102 with native OOB client, a central IoT gateway monitor 104, edge IoT gateways 106 in same subnet (with OOB application), an IoT hub 108, an IoT OOB registration and device update service 110, IoT application and backend services 112, a cloud IoT platform 114, internet 116, an organization IT network 118, a sensor 120, a docker application container 122 that includes a Device Communication Layer (DCL) 124 and a Protocol Translation Layer (PTL) 126, Edge SDK (Software Development Kit) and run time 128, and a native OOB application 130. The native OOB application 130 further includes a P2P client 132, a device

update service 134, a logger 136, a peer health monitor 138, a system health monitor 140, and an OOB client 142.

[017] The IoT hub 108 may be communicatively coupled to the IoT OOB registration and device update service 110 and the IoT application and backend services 112 on the cloud IoT platform 114 via, the internet 116. In accordance with an embodiment, the central IoT gateway monitor 104 may be communicatively coupled to the edge IoT gateway 102 and the edge IoT gateways 106, via the organization IT network 118. The OOB client 142 of the edge IoT gateway 102 may be communicatively coupled to the IoT OOB registration and the device update service 110 on the cloud IoT platform 114, via the internet 116. The Edge SDK and run time 128 may be communicatively coupled to the IoT Hub 108 on the cloud IoT platform 114, via the internet 116. Further, the edge IoT gateways 106 may be communicatively coupled to the IoT OOB registration and device update service 110 on the cloud IoT platform 114, via the internet 116. The edge IoT gateways 106 may be communicatively coupled to the IoT Hub 108 on the cloud IoT platform 114, via the internet 116.

[018] In accordance with an embodiment, the organization IT network 118 may correspond to any communication network that includes a communication medium through which the edge IoT gateway 102 with native OOB client and the edge IoT gateways 106 in same subnet (with OOB application) may communicate with each other. Examples of such communication network may include, but are not limited to, the Internet, a cloud network, a Wireless Fidelity (Wi-Fi) network, a Personal Area Network (PAN), a Local Area Network (LAN), or a Metropolitan Area Network (MAN). Various devices in the block diagram 100 may be configured to connect to such communication network, in accordance with various wired and wireless communication protocols. Examples of such wired and wireless communication protocols may include, but are not limited to, a Transmission Control Protocol and Internet Protocol (TCP/IP), User Datagram Protocol (UDP), Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), Zig Bee, EDGE, IEEE 802.11, light fidelity(Li-Fi), 802.16, IEEE 802.11s, IEEE 802.11g, multi-hop communication, wireless access point (AP), device to device communication, cellular communication protocols, and Bluetooth (BT) communication protocols.

[019] In accordance with an embodiment, any communication network may be used instead of the internet 116 as a communication medium through which the IoT hub 108, the IoT OOB registration and device update service 110, and IoT application and backend services 112 may

communicate with each other on the cloud IoT platform 114, without a deviation from the scope of the disclosure.

[020] The edge IoT gateway 102 may include suitable logic, circuitry, interfaces, and/or code that may be configured to be deployed at edge, near sensors (such as the sensor 120) or directly connected to the sensors (such as the sensor 120) for data collection from the sensors and use multiple protocols.

[021] The edge IoT gateway 102 and the sensors 120 may be deployed remotely and are connected to IoT application and backend services 112 deployed in a remote cloud (such as, the cloud IoT platform 114). Conventionally, edge devices may be prone to software and hardware failures in working lifetime and are also targets for physical and software hacking. In contrast to conventional mechanisms, with an Out of band (OOB) method of communication with IoT devices, the edge IoT gateway 102 may be connected to a centrally located edge control system (such as, the central IoT gateway monitor 104) which is within a secure confines of an industrial IT network (such as, the organization IT network 118).

[022] For multi-protocol support, the DCL 124 of the edge IoT gateway 102 may be used as an application module to provide multiple industry supported protocol spanned across multiple domains. The DCL 124 may use the protocol plugin architecture where any protocol support can be extended by adding a new plugin, without or with minimum changes to application. For operation, the DCL 124 may require a configuration JSON (JavaScript Object Notation) which defines communication parameters with end devices. JSON may correspond to a lightweight data-interchange format to send data between computers or end devices. JSON may perform reading of data and to convert into a JSON message object and to transfer the data to an output source. JSON may provide writing of data (Control) to the end devices based on the commands received from cloud services input sources. Examples of a list of protocols supported by the DCL 124 are, without limitation: for industrial automation ModBus-RTU, ModBus-TCP, and OPCUA, for smart home use cases: ZigBee, Wi-Fi, BLE, and ZWave may be supported, and for building automation: BACnet, and Modbus may be supported.

[023] The PTL 126 of the edge IoT gateway 102 may be configured to have a dual role of fetching configuration JSON from the cloud services and parsing the input JSON and transforming to DCL 124 understandable configuration JSON. In accordance with an embodiment, the PTL 128 may be configured to support receiving configuration inputs from

two sources, that is, a configuration input from module twin (such as, the IoT Hub 108) and a configuration input from OOB cloud services, received via native edge sidecar application.

[024] The native OOB application 130 may be non-containerized. The P2P client 132, the device update service 134, the logger 136, the peer health monitor 138, the system health monitor 140, and the OOB client 142 of the native OOB application 130 may be executed to maintain the connection to the cloud counterpart of the IoT OOB registration service 110.

[025] The P2P client 132 and the peer health monitor 138 of the native OOB application 130 may be used for certain use cases. One of the use cases is explained in conjunction with FIG. 2. For example, in a factory floor, all the edge IoT gateways may not be a source of data. A mesh network may be created among the edge IoT gateways for P2P communication. One of the edge IoT gateways 102 acting as a master may collect data (such as, log data) from other edge IoT gateways 106 acting as slaves, for uploading client data, thereby making IT configuration much simpler. Therefore, instead of transmitting log data from the other edge IoT gateways to the central IoT gateway monitor 104, the edge IoT gateway 102 acting as a master may collect the log data from the other edge IoT gateways 106 and then transmit the same to the central IoT gateway monitor 104. It is to be noted that the edge IoT gateway 102 and the other edge IoT gateways 106 are OOB clients. Similarly, the peer health monitor 138 may be configured to collect peer health data in real time or near real time from the peer edge IoT gateways of an edge IoT gateway.

[026] The device update service 134 may be configured to update firmware on the edge IoT gateway 102. The firmware may be updated on the edge IoT gateway 102 via the communication path of the IoT OOB registration service 110. The system health monitor 140 of the native OOB application 130 may be configured to send system health data in real time or near real time to the central IoT gateway monitor 104. The OOB client 142 of the native OOB application 130 may be configured to send client logs in real time or near real time to the central IoT gateway monitor 104.

[027] The device update service 134 includes device service management and device update management. The device service management may provide support for control of system services using system daemons. Operations supported by the device service management are start, stop, and restart of system services. Device metrics service may read the device operational parameters like CPU usage, RAM usage, disk usage, number of process running,

number of users logged in and sends to cloud OOB services. The device update management for sidecar application provides device update management services to update components on edge devices Over the Air (OTA) using the cloud OOB services. The device service management may support reception of OTA update trigger, perform OTA update by downloading the update file and apply it according to update type received, and send response of OTA operation success or failure status to the cloud OOB services.

[028] The device service management may further support on failure of new update file, rollback to previous working configurations, for an IoT Edge device, the list of components to update may include, without limitation, device firmware. Operating System (OS) libraries, Container engine (such as, Moby), IoT Edge, CA certificates, and OS (Assuming gateway OS distribution provides patch updates).

[029] The logger 136 of the native OOB application 130 may be configured to perform logger functionalities, such as, but not limited to, log data, push data to cloud, logger initialization and terminate. In accordance with an embodiment, expected data for logger functionalities are that the Docker stores such logs under /var/lib/docker/containers/container-ID/container-ID-json.log on a host machine.

[030] Expected data for logger functionalities may further include log-collector container (like Logstash) to collect logs and send the logs to a log-aggregator (like ElasticSearch with Kibana Dashboard). Expected data for logger functionalities may further include logging, come across different types of log data: database logs, network logs, API logs, front-end app logs, and security logs, and event logs. Since containers may run on the same instances, a shared volume may be used to read and write logs.

[031] The Edge SDK and runtime 128 of the edge IoT gateway 102 may own communication on both sides of the spectrum, that is, devices and cloud. The Edge SDK and runtime 128 may be installed as a native binary on the target OS, such as, Raspbian, Debian, Ubuntu, CentOS, and Microsoft Windows. IoT Edge Runtime of the Edge SDK and runtime 128 may be configured to run as a daemon within the OS, interfaces with the container engine (such as, Moby) to manage the lifecycle of docker application container 122 deployed as a module.

[032] The IoT Edge Runtime of the Edge SDK and runtime 128 may include an IoT Edge Agent as a component that runs as a container. The IoT Edge Agent may be configured to

bootstrap itself each time an edge device is powered on. The IoT Edge Agent may be responsible for downloading the deployment manifest from the cloud IoT platform 114 and maintaining a desired state of configuration of the edge device. The IoT Edge Agent may be configured to pull container images from registries and run the container images, based on a predefined configuration. The IoT Edge Agent may be configured to ensure that the state and configuration of containers are matching an original definition associated with the edge device. In a certain scenario, when a new module is added to the manifest through an IoT Portal (such as, Azure), the IoT Edge Agent may pull the image as soon as the IoT Edge Agent discovers the change. Similarly, the IoT Edge Agent may be configured to terminate the containers when the containers are no more a part of the manifest. In accordance with an embodiment, the IoT Edge Agent may be configured to manage the interaction between the cloud IoT platform 114 and a local runtime on the edge IoT gateway 102 to maintain the desired state.

[033] The second component of IoT Edge Runtime may correspond to an edge hub that mimics the IoT Hub 108 in the public cloud or the cloud IoT platform 114. The edge hub may be configured to provide offline capabilities of the IoT Hub 108 by exposing authentication and communication services to leaf devices, such as, the edge IoT gateways 106 in same subnet (with OOB application). A module representing a device may have logic to get authenticated with the local hub (or edge hub). Similarly, the module representing a device can send telemetry data to the edge hub that will forward the telemetry data to the upstream components which are other modules defined as a part of the manifest.

[034] In accordance with an embodiment, the edge hub may be configured to expose same API as its public cloud counterpart, that is, the IoT hub 108. Such a design of the edge hub may reduce the effort required to refactor devices for the edge. Since the edge hub may cache the credentials after the runtime, the edge hub gets authenticated during a handshake with the IoT Hub 108 in the cloud IoT platform 114. The edge hub may act as a communication broker facilitating local device communication. The edge hub may be configured to support standard protocols of the IoT Hub 108. Such standard protocols may include, but not limited to, Advanced Message Queuing Protocol (AMQP), Message Queue Telemetry Transport (MQTT), and Hypertext Transfer Protocol (HTTP). In accordance with an embodiment, modules may be placed close to the devices while the edge agent and the edge hub may be placed close to the control plane of the cloud. The runtime manages and orchestrates the workflow involved in connecting the ends of the spectrum, that is, devices and the cloud.

[035] The central IoT gateway monitor 104 may include suitable logic, circuitry, interfaces, and/or code that may be configured to have the OOB with a separate communication channel and a centrally located logs and operational monitoring system which reads logs, delivers software updates and collects system health information of the devices at regular intervals. As a result, the central IoT gateway monitor 104 facilitates fail-safe operation, security and regular software updates. Operational monitoring of IoT devices (such as, the edge IoT gateway 102 and the sensor 120) in the edge network (such as, the organization IT network 118) may have benefits of fast response time, low internet and cloud usage costs, secure IT management because of no exposure of gateway details of software and hardware configuration, triangulation of failure through integration with peer-to-peer monitoring and use of network management tools.

[036] The central IoT gateway monitor 104 may be responsible for collecting log information and providing real time diagnostics to an operator of an IoT network in the organization IT network 118. By way of an example, the central IoT gateway monitor 104 uses a functionality from loggers of all IoT gateways to collect and display interactive log and health data onto Elastic Search with Kibana Dashboard. In accordance with an embodiment, the central IoT gateway monitor 104 may also have its own database to collect historical performance data for machine learning (ML) and artificial intelligence (AI) analysis of failures of gateways and production systems.

[037] The central IoT gateway monitor 104 may be configured to integrate and enrich data from Manufacturing Execution System (MES) and Supervisory control and data acquisition (SCADA) systems for reliable data for control and analysis.

[038] FIG. 2 is a block diagram that illustrates an exemplary OOB system topology for Peer to Peer (P2P) communication for collecting and aggregating data for rapid on-premise decision making with service mesh, in accordance with an embodiment of the disclosure. FIG. 2 is explained in conjunction with elements from FIG. 1.

[039] With reference to FIG. 2, there is shown a block diagram 200 of the exemplary OOB system. There is further shown a gateway mesh of edge IoT gateways, viz., A, B, and C with a respective sensor communicatively coupled to each of the edge IoT gateways, viz., A, B, and C. In accordance with an embodiment, the edge IoT gateways A, B, and C may have OOB P2P communication with each other for data, such as, but not limited to, sensor data and logging

data. The P2P communication is an advantage of the OOB native application of each of the edge IoT gateways, viz., A, B, and C.

[040] The edge runtime and application module containers of the edge IoT gateways, viz., A, B, and C are orchestrated from the IoT platform environment in the cloud. For example, the edge IoT gateway A does not know of other peer edge IoT gateways B and C in the same network. However, it is important for use cases, like production line monitoring, where a real-time data is collected from sensors and edge IoT gateways A, B, and C placed at different locations in a production line and such real time data needs to be acted upon immediately in case of a failure or anomaly in the production line.

[041] Conventionally, IoT systems are not part of a control system, however the IoT systems may be required to respond in real time for predictive data alerts as they would implement the ML and AI algorithms necessary for failure detection. In contrast to the conventional IoT systems, a P2P peer client module running on native OS of the edge IoT gateway (such as, A) may have the ability to connect to other edge IoT gateways (such as, B and C) in the organization IT network 118 over HTTP Representational State Transfer (ReST) Application Programming Interface (API) or direct TCP client socket communication methods.

[042] The P2P client (such as, the P2P client of the edge IoT gateway A) may discover other gateways (such as, the P2P client of the edge IoT gateway B and the P2P client of the edge IoT gateway C) with the help of the central IoT gateway monitor 104 which has information on a local network topology and maintains the connections with each of the edge IoT gateways (such as, the edge IoT gateways A, B, and C) in the local network (such as, the organization IT network 118).

[043] A gateway implementing the role of a master gateway in the production line (as shown in FIG. 2), may request the central IoT gateway 104 for peer gateway IP address and connect to those gateways using P2P client. By way of an example, the P2P client in the edge IoT gateway A will connect to the edge IoT gateway B and the edge IoT gateway C to collect data from them in real time. Hence, this may create a gateway mesh communication path between the 3 gateways, viz. the edge IoT gateways A, B, and C. In accordance with an embodiment, the edge IoT gateway A may be configured as “master” during the IoT gateway OOB registration and provisioning process at the IoT OOB registration service 110.

[044] In accordance with an embodiment, the P2P client in the master (that is, the edge IoT gateway A) may receive data and also provide the data to the IoT client application running the algorithms in the docker container. The P2P client data may appear as another sensor to the IoT application in the docker container.

[045] Further the gateway IoT application associated with the edge IoT gateway may process data on the edge IoT gateway or send the data to IoT application in cloud as per the implementation. Reliability of data may be improved by comparing contextual and complimentary data from other gateways and their sensors. Redundancy may be provided by virtue of multiple sensor input from other IoT gateways deployed in an industrial premise to collect localized data or data on the same machinery.

[046] FIGs. 3A-3B collectively illustrate an exemplary registration call flowchart for IoT Gateway and Central IoT gateway monitor within premises of organization IT network, with the IoT platform in cloud, in accordance with an embodiment of the disclosure. FIGs. 3A-3B are explained in conjunction with elements from FIG. 1 to FIG. 2. With reference to FIG. 3A, there is shown a flowchart 300A. The operations of the flowchart 300A may start at 302 and proceed to 304.

[047] At 302, IoT gateways may be registered with the IoT hub 108. In accordance with an embodiment, the edge IoT gateway 102 and the edge IoT gateways 106 may register with the IoT hub 108. In accordance with an embodiment, one or more of the edge IoT gateway 102 and the edge IoT gateways 106 may be registered with the IoT hub 108. An example of the IoT hub 108, without limitation is Azure IoT hub.

[048] At 304, the IoT gateways may be registered with the IoT OOB registration and device update service 110. In accordance with an embodiment, the edge IoT gateway 102 and the edge IoT gateways 106 may register with the IoT OOB registration and device update service 110. In accordance with an embodiment, one or more of the edge IoT gateway 102 and the edge IoT gateways 106 may be registered with the IoT OOB registration and device update service 110. The registration of the IoT gateways with the IoT OOB registration and device update service 110 may be parallel to the registration of the IoT gateways with the IoT hub 108 to manage peer to peer (P2P) network among the IoT gateways.

[049] At 306, the central IoT gateway health monitor 104 may be registered. In accordance with an embodiment, the central IoT gateway health monitor 104 may register with the IoT OOB registration and device update service 110.

[050] At 308, OOB client list may be downloaded. In accordance with an embodiment, the central IoT gateway health monitor 104 may download the OOB client list from the IoT OOB registration and device update service 110. Therefore, the central IoT gateway health monitor 104 may receive information of all the IoT gateways in the enclosed network (such as, the organization IT network) of the central IoT gateway health monitor 104.

[051] The registration with the IoT OOB registration and device update service 110 may add new features, such as, firmware updates in a secure manner, binaries of an IoT gateway will not be exposed out of a secure organization IT network 118, and a cloud agnostic registration service. By way of an example, the IoT OOB registration and device update service 110 can be deployed on any cloud platform, such as, but not limited to, Amazon Web Services (AWS), Google Cloud, and Azure and still be used as a common mechanism for firmware update and managing local gateway information in a secure manner.

[052] At 310, OOB client list may be updated. In accordance with an embodiment, the IoT OOB registration and device update service 110 may update the OOB client list on the central IoT gateway health monitor 104. Thereafter, the OOB client list may be maintained by the central IoT gateway health monitor 104 in the secure organization IT network 118. Advantageously, the central IoT gateway health monitor 104 need not connect with any cloud platform, such as, Azure cloud platform for gateway information (or client list) where the chances of security compromise are high.

[053] At 312, local OOB clients may be connected. In accordance with an embodiment, after receiving client list update, the IOT gateways, such as the edge IoT gateway 102 may connect with local OOB clients from the local OOB client list received from the central IoT gateway health monitor 104. In accordance with an embodiment, the edge IoT gateway 102 may start collecting data, such as, log data in real time or near real time from other edge IoT gateways 106.

[054] At 314, real time client logs and system health may be received. In accordance with an embodiment, the central IoT gateway health monitor 104 may receive the real time client logs and the system health from the edge IoT gateway 102 and the edge IoT gateways 106.

[055] At 316, optionally, on demand OOB data for cloud dashboards may be exchanged between the central IoT gateway health monitor 104 and the IoT OOB registration and device update service 110. By way of an example, a management person of an organization wants to fetch log data from the edge IoT gateways and the management person is not a part of the organization IT network 118. The management person can fetch the log data on need basis from the central IoT gateway health monitor 104 using cloud dashboard. The management person can fetch the log data on need basis without receiving huge amount of data in real time in a sure manner pertaining to optional IT policy.

[056] With reference to FIG. 3B, there is shown a flowchart 300B. The operations of the flowchart 300B may start at 318 and proceed to 320.

[057] At 318, IoT gateways may be registered with the IoT hub 108. In accordance with an embodiment, the edge IoT gateway 102 and the edge IoT gateways 106 may register with the IoT hub 108. In accordance with an embodiment, one or more of the edge IoT gateway 102 and the edge IoT gateways 106 may be registered with the IoT hub 108.

[058] At 320, IoT gateways may be registered with IoT OOB registration and device update service 110. In accordance with an embodiment, the edge IoT gateway 102 and the edge IoT gateways 106 may register with the IoT OOB registration and device update service 110.

[059] At 322, the central IoT gateway health monitor 104 may register with the IoT OOB registration and device update service 110.

[060] At 324, registered IoT gateway list may be downloaded. In accordance with an embodiment, the central IoT gateway health monitor 104 may download registered IoT gateway list from the IoT OOB registration and device update service 110.

[061] At 326, the IoT OOB registration and device update service 110 may transmit registered gateway list update to the central IoT gateway health monitor 104.

[062] At 328, the central IoT gateway health monitor 104 may connect to the local OOB clients on the edge IoT gateway 102 and the edge IoT gateways 106.

[063] At 330, obtain real time client logs and system health may be obtained. In accordance with an embodiment, the central IoT gateway health monitor 104 may obtain real time client logs and system health from the edge IoT gateway 102 and the edge IoT gateways 106.

[064] At 332, the IoT OOB registration and device update service 110 may transmit multi-protocol configuration data to the edge IoT gateway 102. In accordance with an embodiment, the edge IoT gateway 102 and the edge IoT gateways 106 may configure DCL and PTL. As explained in FIG. 1, the edge IoT gateway 102 includes the docker application container 122 that connects with the sensor 120. The DCL and PTL may be configured via the OOB registration and device update service 110. The native OOB application 130 may command the OOB registration and device update service 110 to configure DCL and PTL only. Therefore, by way of an example, the edge IoT gateway 102 may be able to collect sensor data and send the same data to the logger 136.

[065] At 334, edge IoT gateway 102 may request the central IoT gateway health monitor 104 for local network topology. In accordance with an embodiment, the edge IoT gateway 102 may need to discover peers in the organization IT network 118. By way of an example, when the edge IoT gateway 102 acts as a master device, the edge IoT gateway 102 may want to know the slave devices of the edge IoT gateway 102. For the same reason, the edge IoT gateway 102 may want to know the configuration of the local network topology.

[066] At 336, the central IoT gateway health monitor 104 may transmit the local network topology information to the edge IoT gateway 102 and the edge IoT gateways 106. In accordance with an embodiment, the edge IoT gateway 102 may download the edge IoT gateway 102 from the central IoT gateway health monitor 104.

[067] Furthermore, one or more computer-readable storage media may be utilized in implementing embodiments consistent with the present disclosure. A computer-readable storage medium refers to any type of physical memory on which information or data readable by a processor may be stored. Thus, a computer-readable storage medium may store instructions for execution by one or more processors, including instructions for causing the processor(s) to perform steps or stages consistent with the embodiments described herein. The term “computer-readable medium” should be understood to include tangible items and exclude carrier waves and transient signals, i.e., be non-transitory. Examples include random access memory (RAM), read-only memory (ROM), volatile memory, nonvolatile memory, hard drives, CD ROMs, DVDs, flash drives, disks, and any other known physical storage media.

[068] It will be appreciated that, for clarity purposes, the above description has described embodiments of the disclosure with reference to different functional units and processors.

However, it will be apparent that any suitable distribution of functionality between different functional units, processors or domains may be used without detracting from the disclosure. For example, functionality illustrated to be performed by separate processors or controllers may be performed by the same processor or controller. Hence, references to specific functional units are only to be seen as references to suitable means for providing the described functionality, rather than indicative of a strict logical or physical structure or organization.

[069] Although the present disclosure has been described in connection with some embodiments, it is not intended to be limited to the specific form set forth herein. Rather, the scope of the present disclosure is limited only by the claims. Additionally, although a feature may appear to be described in connection with particular embodiments, one skilled in the art would recognize that various features of the described embodiments may be combined in accordance with the disclosure.

[070] Furthermore, although individually listed, a plurality of means, elements or process steps may be implemented by, for example, a single unit or processor. Additionally, although individual features may be included in different claims, these may possibly be advantageously combined, and the inclusion in different claims does not imply that a combination of features is not feasible and/or advantageous. Also, the inclusion of a feature in one category of claims does not imply a limitation to this category, but rather the feature may be equally applicable to other claim categories, as appropriate.

WE CLAIM:

1. An Out of Band (OOB) method for operation management of a set of Internet of Things (IoT) gateways deployed at edge, the OOB method comprising:

registering at least one IoT gateway from the set of IoT gateways with an OOB IoT registration service, wherein a central gateway health monitor is registered with the OOB IoT registration service to obtain OOB client list associated with the set of IoT gateways, and wherein the OOB IoT registration service manages gateway-to-gateway communication via a local communication network;

upon registration, connecting the at least one IoT gateway from the set of IoT gateways with the central gateway health monitor, wherein the central gateway health monitor is configured to have OOB management of the set of IoT gateways using a separate communication channel for centrally located logs associated with at least one of the set of IoT gateways and an operating monitoring system for the set of IoT gateways;

obtaining an OOB client list at the at least one IoT gateway from the central gateway health monitor, wherein the OOB client list corresponds to a list of IoT gateways registered with the OOB IoT registration service;

connecting the at least one IoT gateway with one or more of the set of IoT gateways, via the local communication network for real time log data and system health data associated with the one or more of the set of IoT gateways, based on the OOB client list; and

transmitting the real time log data and system health data associated with the one or more of the set of IoT gateways from the at least one IoT gateway to the central gateway health monitor for the operation management of the set of IoT gateways at the edge.

2. The OOB method as claimed in claim 1, further comprises:

obtaining, at the at least one IoT gateway, multi-protocol configuration data associated with each of the set of IoT gateways from the OOB IoT registration service, wherein the multi-protocol configuration data is indicative of configuration of Device Communication Layer (DCL) and Protocol translation Layer (PTL) on a native OOB application of the at least one IoT gateway.

3. The OOB method as claimed in claim 2, further comprises:

obtaining, at the at least one IoT gateway, local communication network topology information from the central gateway health monitor, based on a response to a request

for the local communication network topology information from the at least one IoT gateway to the central gateway health monitor;

determining, at the at least one IoT gateway, a master device and one or more slave devices corresponding to each of the set of IoT gateways, based on the local communication network topology information;

establishing a gateway-to-gateway communication among the at least one IoT gateway and one or more of the set of IoT gateways; and

aggregating a local gateway-to-gateway communication data at the edge, based on the established gateway-to-gateway communication.

4. The OOB method as claimed in claim 1, wherein the OOB IoT registration service is further configured to perform firmware update of the at least one IoT gateway.

5. The OOB method as claimed in claim 1, further comprises:

registering at least one IoT gateway from the set of IoT gateways with an IoT hub, wherein the IoT hub is configured to manage the set of IoT gateways via internet.

Dated this 3rd day of Aug 2022

-- Digitally Signed--
Bhanu Prasad (INPA No: **3253**)
Manager, IPR Dept.,
L&T Technology Services Limited,
DLF 3rd Block, 2nd Floor,
Manapakkam, Chennai - 600089.

ABSTRACT

OUT OF BAND (OOB) SYSTEM AND METHOD FOR MANAGING INTERNET OF THINGS (IoT) GATEWAYS

OOB method for operation management of a set of IoT gateways is provided. The OOB method includes registering at least one IoT gateway with IoT hub and an OOB IoT registration service. A central gateway health monitor is also registered with OOB IoT registration service to obtain OOB client list associated with set of IoT gateways. The OOB method includes connecting at least one IoT gateway with central gateway health monitor configured to have OOB management of set of IoT gateways using separate communication channel for centrally located logs and operating monitoring system for the set of IoT gateways. The OOB method includes obtaining OOB client list, connecting at least one IoT gateway with one or more of set of IoT gateways, via local communication network and transmitting data from at least one IoT gateway to central gateway health monitor for operation management of set of IoT gateways at edge.

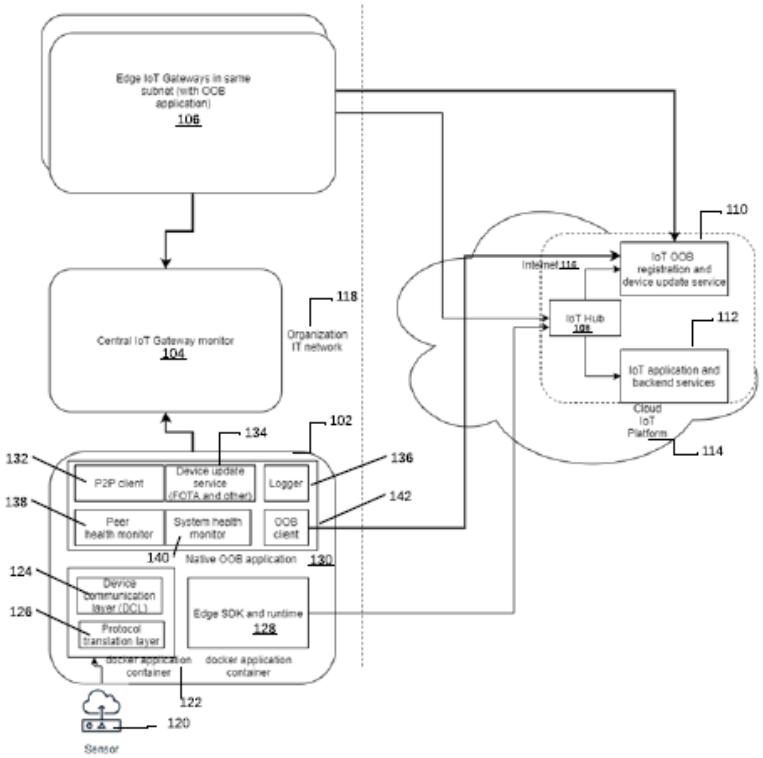

Fig 1

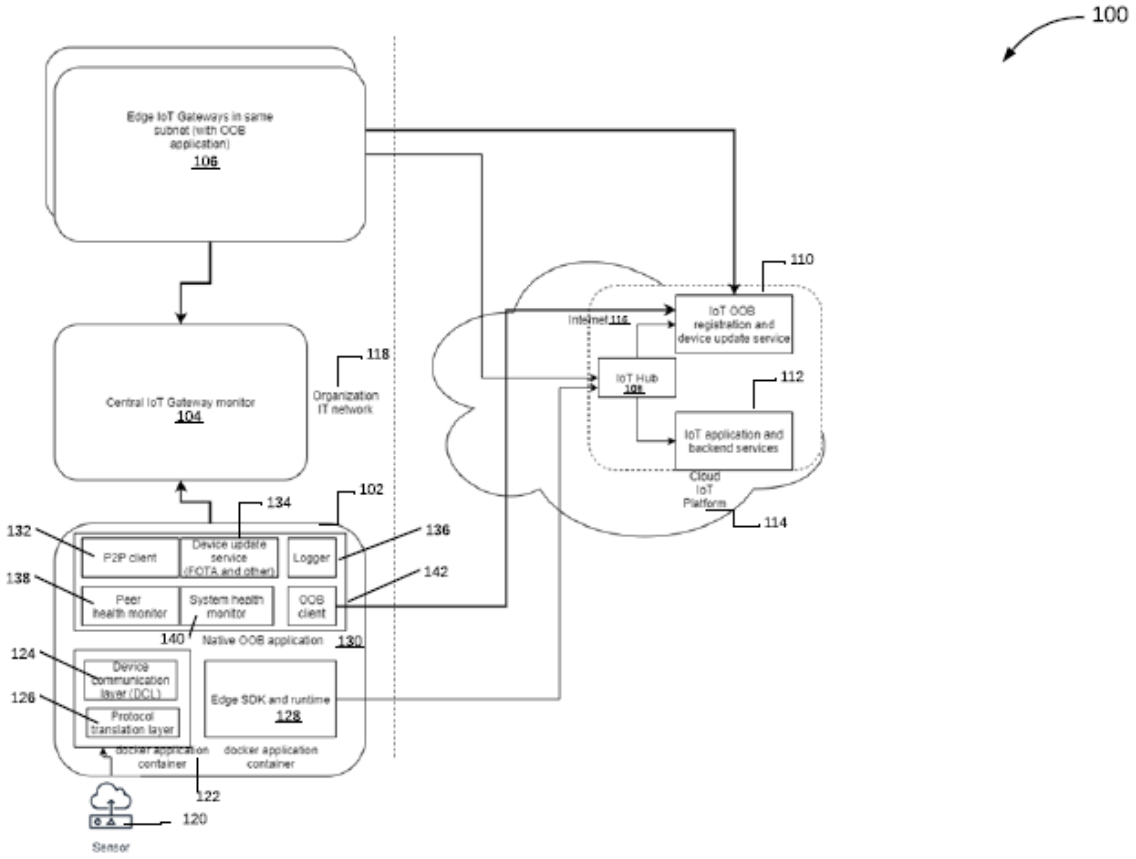

Fig 1

-- Digitally Signed--
Bhanu Prasad (INPA No: 3253)
Manager, IPR Dept.,
L&T Technology Services Limited,
DLF 3rd Block, 2nd Floor,
Manapakkam, Chennai - 600089.

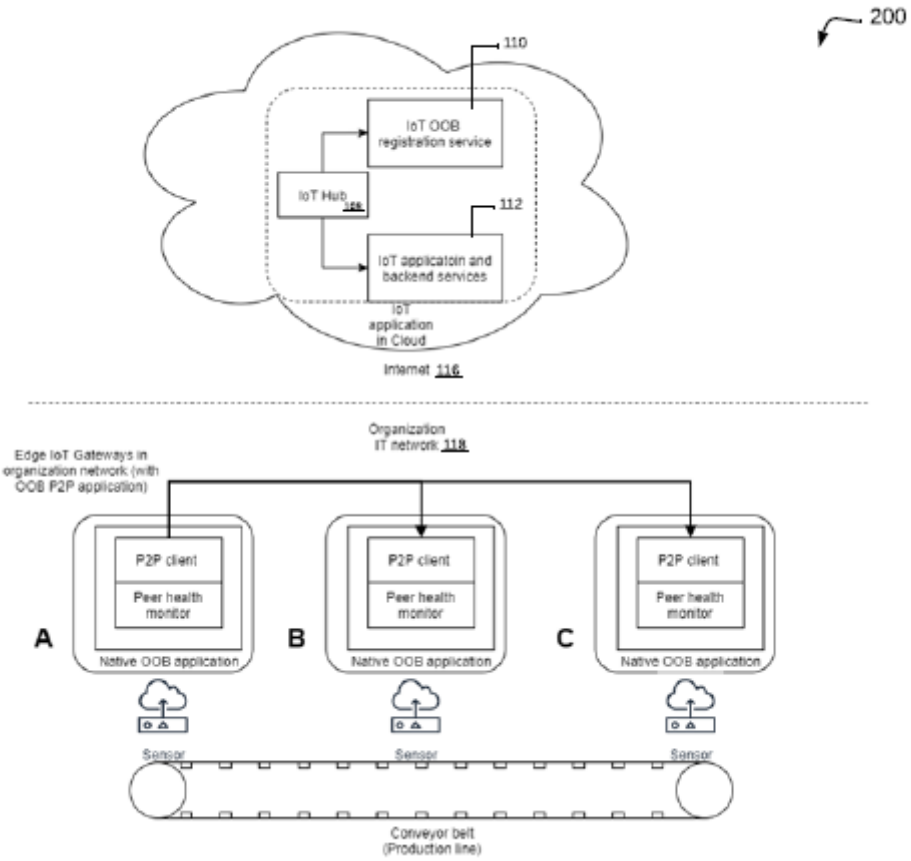

Fig 2

-- Digitally Signed--
Bhanu Prasad (INPA No: 3253)
Manager, IPR Dept.,
L&T Technology Services Limited,
DLF 3rd Block, 2nd Floor,
Manapakkam, Chennai - 600089.

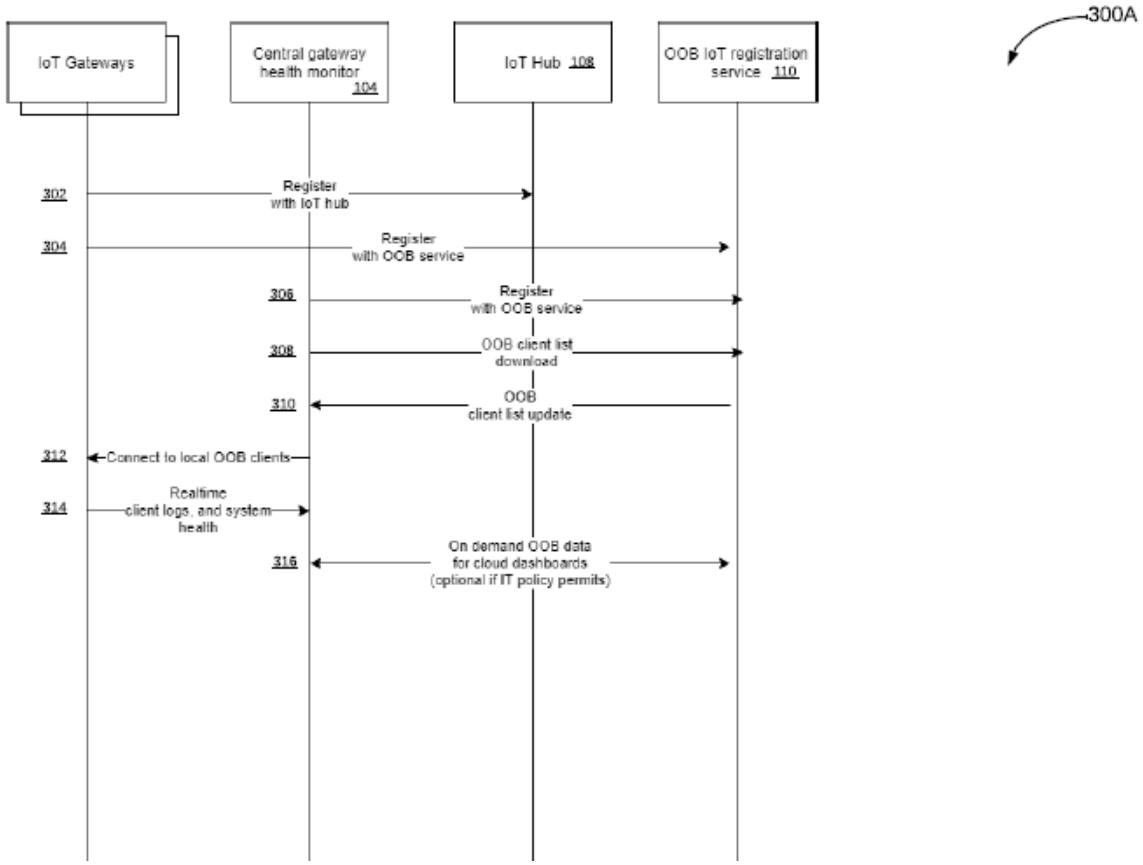

Fig 3a

-- Digitally Signed--
Bhanu Prasad (INPA No: 3253)
Manager, IPR Dept.,
L&T Technology Services Limited,
DLF 3rd Block, 2nd Floor,
Manapakkam, Chennai - 600089.

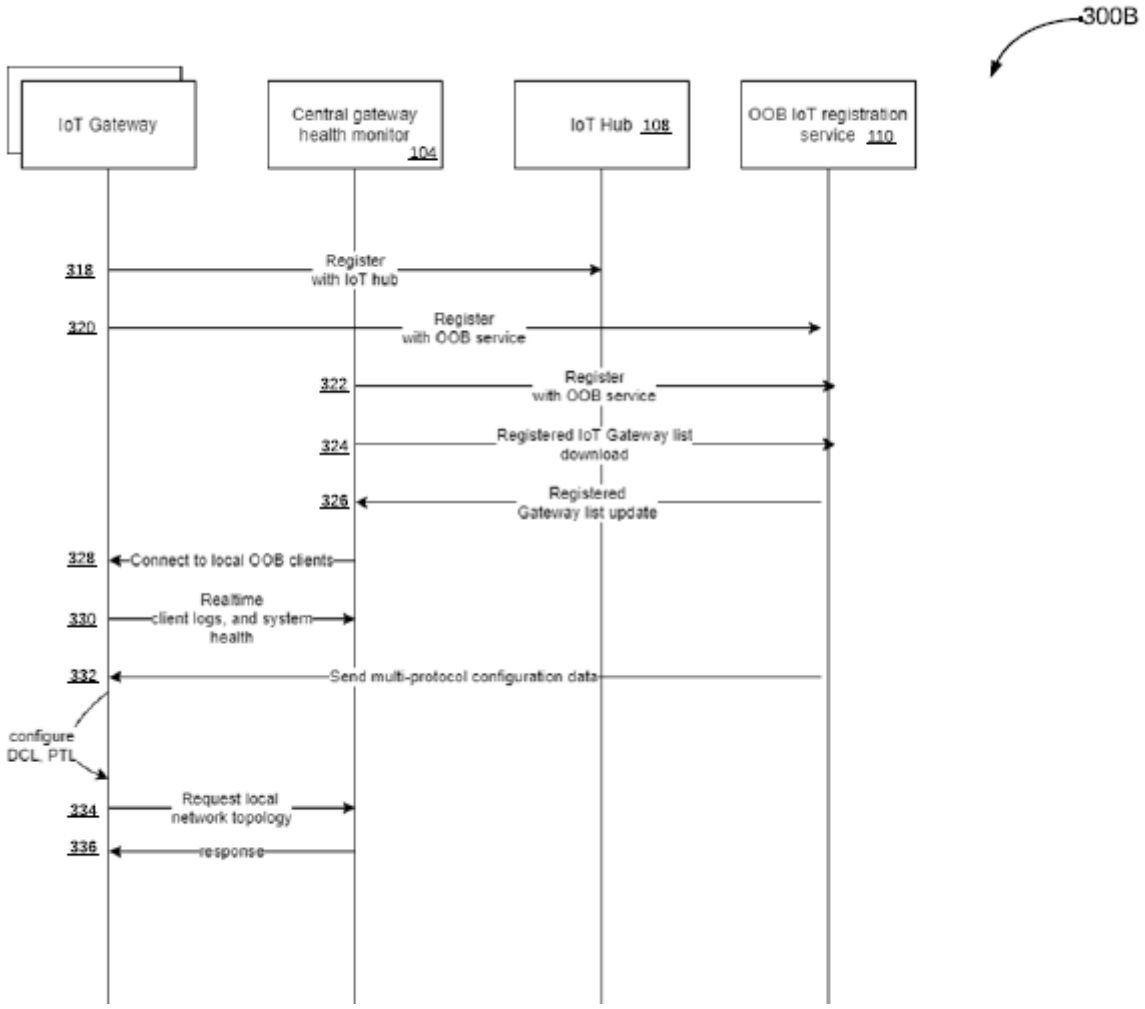

Fig 3b

-- Digitally Signed--
Bhanu Prasad (INPA No: 3253)
Manager, IPR Dept.,
L&T Technology Services Limited,
DLF 3rd Block, 2nd Floor,
Manapakkam, Chennai - 600089.