short-range, typically a meter or less because the physical mechanism is based on induction rather than backscatter.

- **Passive RFID:** Passive RFID tags does not have their own power source. It uses power from the reader. In this device, RF tags are not attached by a power supply and passive RF tag stored their power. When it is emitted from active antennas and the RF tag are used specific frequency like 125-134KHZ as low frequency, 13.56MHZ as a high frequency and 856MHZ to 960MHZ as ultra-high frequency.
    - No need embedded power
    - Tracking inventory
    - Has unique identification number
    - Sensitive for interference
    - Semi-passive RFID

- **Active RFID:** In this device, RF tags are attached by a power supply that emits a signal and there is an antenna which receives the data. means, active tag uses a power source like battery. It has it's own power source, does not require power from source/reader.
    - Embedded power: communication over large distance
    - Has unique identifier /identification number
    - Use other devices like sensors
    - Better than passive tags in the presence of metal

There are also other forms of RFID using other frequencies, such as LF RFID (Low-Frequency RFID), which was developed before HF RFID and used for tracking.

BACnet Protocols

BACnet, which stands for "Building Automation and Control Networks," is a widely used communication protocol specifically designed for building automation and control systems. It enables interoperability and communication between various devices and systems in buildings, such as heating, ventilation, air conditioning (HVAC), lighting, access control, fire detection, and more. BACnet is essential for creating integrated and efficient building automation systems.

Here are some key features and details about the BACnet protocol:

Open Standard: BACnet is an open, vendor-neutral protocol standardized under the ANSI/ASHRAE (American Society of Heating, Refrigerating and Air-Conditioning

Engineers) 135 standard. This openness promotes competition, innovation, and compatibility among different manufacturers' products.

Interoperability: BACnet is designed to ensure that devices from various manufacturers can communicate and work together seamlessly within a building automation system. This is crucial for avoiding vendor lock-in and enabling flexible system design.

Client-Server Architecture: BACnet follows a client-server communication model. Devices in a BACnet network can act as clients (requesting data or actions) or servers (providing data or services). This model allows for efficient data exchange and control.

Data Objects: BACnet defines standardized data objects that represent various aspects of building equipment and systems. These objects include temperature sensors, valves, lights, alarms, schedules, and more. Devices communicate by reading and writing data to these objects.

Protocols and Network Options: BACnet supports various data link and network layers, including Ethernet, RS-485, and IP (Internet Protocol). This flexibility allows it to operate over different types of networks, making it suitable for both wired and wireless building automation systems.

Data Types: BACnet defines a wide range of data types to accommodate different types of information, such as binary, analog, multi-state, and character string data. This versatility makes it adaptable to diverse building systems.

Routing and Segmentation: BACnet networks can be segmented into different parts, and devices can route messages to their intended destinations. This feature is important for scaling BACnet networks to larger buildings or complex systems.

Alarm and Event Handling: BACnet includes support for alarms and events, allowing devices to report and respond to abnormal conditions, such as temperature spikes or equipment failures.

Scheduling: BACnet supports scheduling, enabling automation of building systems based on time and day-of-week schedules. This is useful for optimizing energy usage and comfort.

Security: BACnet includes security features to protect communication and data integrity. It supports authentication, encryption, and access control to safeguard building automation systems.

Integration with Other Protocols: BACnet can be integrated with other communication protocols, such as Modbus or LonWorks, to extend interoperability to legacy systems or devices that use different standards.

Global Adoption: BACnet is used worldwide and has been adopted by a wide range of industries, including commercial buildings, industrial facilities, hospitals, and residential complexes.

BACnet plays a crucial role in modern building automation, enabling efficient and intelligent control of various building systems for energy efficiency, comfort, and safety. Its open, standardized nature promotes flexibility and innovation in the development of building automation solutions.

Communications protocol for Building Automation and Control (BAC) networks. Provides mechanisms for computerized building automation devices to exchange information. Designed to allow communication of building automation & control system for application like Heating, Ventilating and Air-conditioning Control (HVAC), Lighting Control, Access Control, Fire Detection Systems and their Associated Equipment.

Defines a number of services that are used to communicate between building devices Protocol services include Who-Is, I-Am, Who-Has, I Have which are used for Device & Object discovery. Services such as Read-Property and WriteProperty are used for data sharing. Defines 60 object types that are acted upon by services.

Building Automation and Control Networks developed by American Society of heating, Refrigerating and Air Conditioning Engineers (ASHRAE). It is a data communication protocol designed for communication between building automated system components. This is an Object Oriented protocol.

- Objects : Physical device, temperature input(analog input) , A relay control(binary output), schedules
- Services : Used to perform read, write and I/O
- BACNet Objects are evaluated and controlled by their properties.

Object Name	"Lighting Area"
Object Type	BINARY_OUTPUT
Present Value	Active
Status_Flags	Normal, In-Service
Out_Of_Service	False
Inactive_Text	"Off"
Active_Text	"On"

BACNet Network Types: Physical and Data Link Layer BACNet IP
- Used with existing ethernet, WAN BACNet MS/TP
- Uses Twisted Pair EIA -485 upto 4000 feet BACNet ISO 8802-3
- Limited to Single Infrastructure without IP routers BACNet P2P
- Used only for dial-up telephone networks

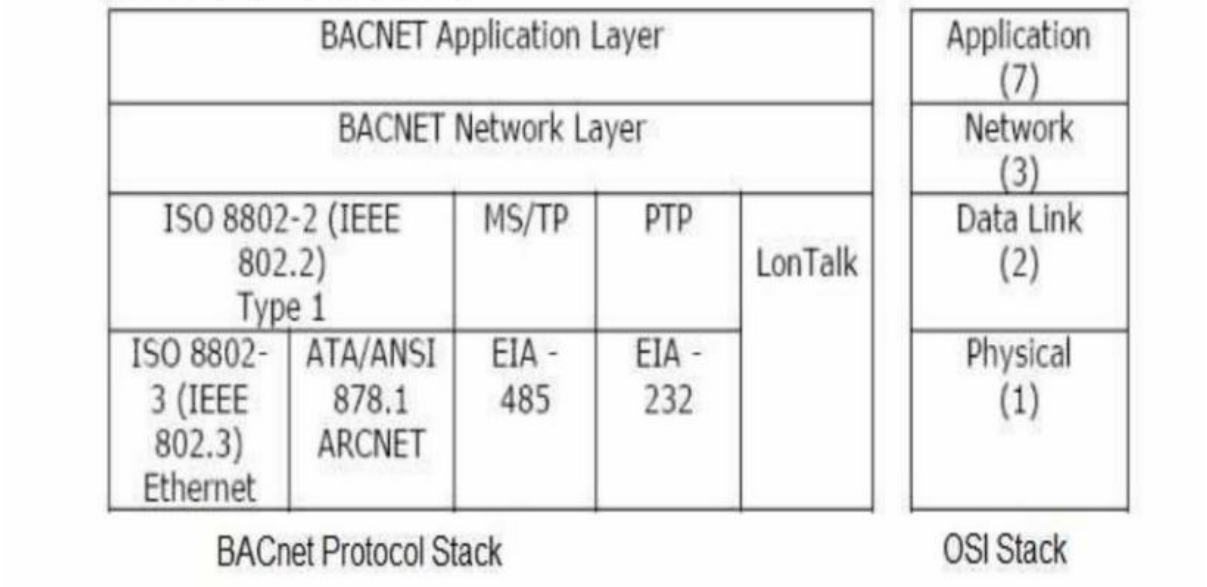
Bacnet Protocol Architecture

BACNet Services

BACNet services are formal requests that one BACNet device sends to another to ask it to do something

Categories :

– Object Access (Read, Write, Create, Delete)
– Alarm and Event (Alarms and Changes of State)
– File Access (Trend data, Data transfer)
– Remote Device Management (Discover, Time Synchronization, Backup and Restore Database, Initialization)
– Virtual Terminal (HMI via menus)
– Who Is, I Am , Who-Has, I-Have



BACnet protocol architecture is a collapsed architecture that matches to 4-layers of the OSI model. The four layers in the BACnet architecture mainly include Application, Network, Data Link & Physical. Even though, just the Network layer & Application layer are simply BACnet. The above architecture is the BACnet protocol stack which includes different layers as shown in the diagram. This protocol is a collapsed version of the OSI stack. The transport and session layers are not used. The application layer takes on the functions of these two layers.

BACnet Physical Layer

The upper layers of BACnet do not depend on the physical layer. So the Physical layer of BACnet makes it feasible for BACnet to be executed on different networks. The physical layers of BACnet have been specified with ARCNET, Ethernet, IP tunnels, BACnet/IP, RS-232, RS485, and Lonworks/LonTalk. RS232 is for point-to-point communication. RS485 supports up to 32 nodes with a distance of 1200 m at 76Kbps.

BACnet Protocol Link Layer

BACnet protocol is implemented directly with LonTalk or IEEE802.2 link layers. So it specifies Point to Point (PTP) data link layer for RS232 connections. It specifies MS/TP data link layer intended for RS-485 connections. The standard simply specifies BVLL (BACnet Virtual Link Layer) which states all the services required through the BACnet device at this link layer.

IP BACnet Virtual Link Layer encapsulates required control data in a header of BACnet virtual link control information. Because of IP, BVLL, and BACnet protocol devices can directly communicate over IP networks without the requirement of any router device.

BACnet protocol utilizes BBMD (BACnet broadcast management device) concept which executes the required broadcast for the preferred link layer. So, the BACnet broadcast message is changed into IP-based broadcast or multicast messages.

BACnet Network Layer

This layer simply specifies the required addresses of the network for routing. BACnet network includes a minimum of one or above segments that are connected with bridges once they utilize similar LAN technologies. If they utilize various LAN protocols then they are connected through routers.

Application Layer

BACnet does not separate presentation as well as application layers. So it takes care of reliability & sequencing or segmentation mechanisms generally connected with both the session & transport layers. BACnet includes devices like objects to exchange service primitives which are described with ASN.1 syntax & serialized with ASN.1 BER.

BACnet Security Layer

The concept of BACnet security can be understood easily with an example say when BACnet device-A requests a session key from the key server for establishing secure communication through device-B, then this key is transmitted to both the device-A & device-B through the key server which is known as ‘SKab’. BACnet protocol uses 56-bit DES encryption.

How Does Bacnet Protocol Work?

BACnet is a typical electronic communication protocol that works by allowing different kinds of manufacturers’ building automation as well as monitoring systems like fire alarms, HVAC, and perimeter security for communicating with each other. This protocol can work with nearly any normal data protocol including TCP/IP.

BACnet protocol enables the comprehensive BMSs (building management systems) development that allows operators to construct, observe & control different building systems within a single application. This protocol is also used to expand the flexibility & scope of the automation that can be executed. For instance, an automation system could be setup such that once the fire protection system notices a fire, then the system sends commands to the following.

- To the control system of the elevator to send all elevators to the ground floor immediately.
- To the paging system of the building to transmit an audible voice signal to inform occupants of the building wherever the blaze was detected & how to go out from the building.
- From the audio or visual systems of the building to flash messages on TV displays within the conference rooms.
- To an interface of phone system for sending alerts through text message to the facilities & engineering teams of the building.

With BACnet protocol, all the data is signified in terms of an object. So each object signifies data regarding a device or component. Signifying information like an object simply provides the benefit that the latest objects can be formed otherwise existing objects can be modified based on the requirements of the user.

An object signifies physical information (physical inputs, outputs) & nonphysical information (software/calculations). It is very significant to note that every object may signify a single portion of information otherwise a group of information which executes the same and exact function.

Advantages

The **advantages of the Bacnet Protocol** include the following.

- BACnet protocol is particularly designed for building automation as well as control networks.
- It doesn’t depend on present LAN or WAN technologies.
- It is an American National Standard & a European pre-standard.
- It is scalable completely from small single building applications to universal networks of devices.
- The implementers of BACnet can securely include non-standard extensions as well as enhancements without influencing existing interoperability.
- It is adopted by the most famous fire protection companies in both the USA & Europe.
- It is supported by different chiller manufacturers like Dunham-Bush, Carrier, McQuay, York & Trane.
- In real building control applications, this protocol has a proven track record.

Disadvantages

The **disadvantages of the Bacnet Protocol** include the following.

The main drawback of the BACnet protocol was a compliant problem. So, because of this issue, the BTL (BACnet Testing Laboratories) was introduced in the year 2000. BTL is compliance & and independent testing organization. The main intention of this is to test the products of BACnet to verify compliance with the standard. Once approved; the product will get the logo of BTL.

The problems or net-worthy attacks which are widely found in this protocol are; Lack of spoofing & authentication, DoS attacks, immobilized network connections, and lack of encryption & write access over devices.

Modbus

MODBUS is an application layer messaging protocol, positioned at level 7 of the OSI model, that provides client/server communication between devices connected on different types of buses or networks.

The industry's serial de facto standard since 1979, MODBUS continues to enable millions of automation devices to communicate. Today, support for the simple and elegant structure of MODBUS continues to grow. The Internet community can access MODBUS at a reserved system port 502 on the TCP/IP stack.

MODBUS is a request/reply protocol and offers services specified by function codes. MODBUS function codes are elements of MODBUS request/reply PDUs. The objective of this document is to describe the function codes used within the framework of MODBUS transactions.

MODBUS is an application layer messaging protocol for client/server communication between devices connected on different types of buses or networks.

Working:

The MODBUS protocol defines a simple protocol data unit (PDU) independent of the underlying communication layers. The mapping of MODBUS protocol on specific buses or network can introduce some additional fields on the application data unit (ADU).

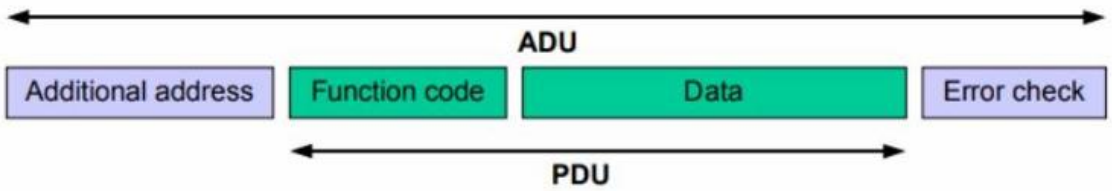


Figure 3: General MODBUS frame

The MODBUS application data unit is built by the client that initiates a MODBUS transaction. The function indicates to the server what kind of action to perform. The MODBUS application protocol establishes the format of a request initiated by a client.

The function code field of a MODBUS data unit is coded in one byte. Valid codes are in the range of 1 ... 255 decimal (the range 128 – 255 is reserved and used for exception responses). When a message is sent from a Client to a Server device the function code field tells the server what kind of action to perform. Function code "0" is not valid. Sub-function codes are added to some function codes to define multiple actions. The data field of messages sent from a client to server devices contains additional information that the server uses to take the action defined by the function code.

This can include items like discrete and register addresses, the quantity of items to be handled, and the count of actual data bytes in the field. The data field may be nonexistent (of zero length) in certain kinds of requests, in this case the server does not require any additional information. The function code alone specifies the action.

If no error occurs related to the MODBUS function requested in a properly received MODBUS ADU the data field of a response from a server to a client contains the data requested. If an error related to the MODBUS function requested occurs, the field contains an exception code that the server application can use to determine the next action to be taken.

For example a client can read the ON / OFF states of a group of discrete outputs or inputs or it can read/write the data contents of a group of registers. When the server responds to the client, it uses the function code field to indicate either a normal (error-free) response or that some kind of error occurred (called an exception response). For a normal response, the server simply echoes to the request the original function code.

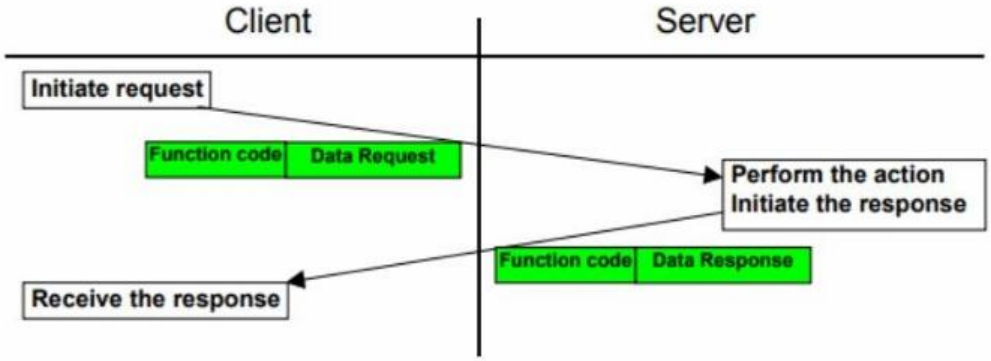


Figure 4: MODBUS transaction (error free)

For an exception response, the server returns a code that is equivalent to the original function code from the request PDU with its most significant bit set to logic 1.

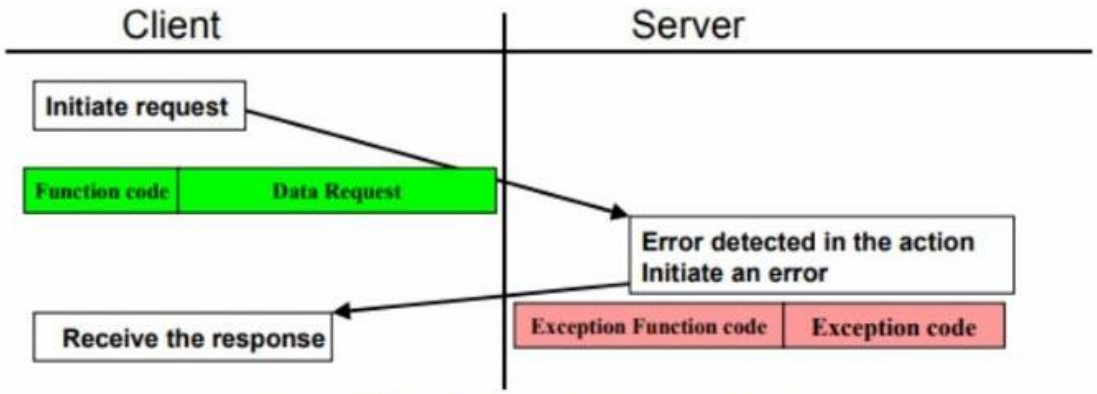


Figure 5: MODBUS transaction (exception response)

KNX

Intelligent bus systems make buildings more cost-effective to operate, safer, more flexible, more energy-efficient and above all more comfortable and convenient. The KNX standard occupies a large share of the market for building automation systems. The KNX building automation system was originally known as the European Installation Bus (EIB), and was developed and marketed by the EIB Association (EIBA).

In 1999, EIBA, Batibus Club International (BCI, France) and the European Home Systems Association (EHSA, Netherlands) amalgamated, the name KNX was adopted, and the Brussels-based KNX Association was set up. The technology used in modern KNX devices is compatible with that of the old EIB system, so all devices bearing either the KNX or the EIB logo are mutually compatible.

The KNX system is a bus system for building control. This means that all devices in a KNX system use the same transmission method and are able to exchange data via a common bus network. This has the following consequences:

- Access to the bus network needs to be clearly regulated (bus access method)
- Most of the data transmitted are not payloads (e. g. light on / light off signals), but address information (i. e. where have the data come from? Where are they going to?)

Another important feature of the KNX bus system is its decentralised structure: there is no need for a central control unit, because the "intelligence" of the system is spread across all of its devices. Centralised units are possible, however, for realising very specialised applications. Every device has its own microprocessor.

A major advantage of KNX's decentralized structure is that, if one device fails, the others continue to function. Only those applications dependent on the failed device will be interrupted. Generally in a KNX system, devices fall into three categories: system devices (power supply, programming interface, etc.), sensors, and actuators. Sensors are devices that detect events in the building (e.g. someone pressing a button, someone moving, a temperature falling above or below a set value, etc.), convert these into telegrams (data packets), and send them along the bus network. Devices that receive telegrams and convert the commands embedded in them into actions are known as actuators. Sensors issue commands, while actuators receive them.

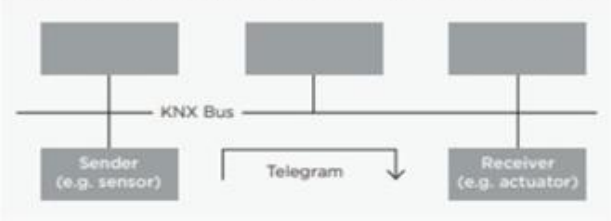


Figure: Sensor/actuator principal

Working Logic:

The key to making a building "intelligent" is to equip it with networked sensors and actuators. There are several different ways of doing this:

(i) Conventional methods

The immediately obvious solution is to employ a star topology, i. e. an arrangement where every socket outlet circuit, ceiling or wall outlet, and light switch is linked by its own (ideally five core) NYM cable to a central distribution board in which the logical relationships are created by contactors, switch relays, and a programmable logic controller (PLC). This works well in reasonably small dwellings. However, the size of the house only needs to increase by a fairly small amount before the extent of the wiring work and size of the power distribution boards required becomes excessive. In a star topology, adding to or extending the system is also very time-consuming in terms of installation and programming.

(ii) Bus technology

A far better solution is to link all sensors and actuators in the building with a "data cable", and enable them to share information with each other (Fig.2). Each device can then communicate with every other device, for example: a light switch can "talk" to a dimmer and tell it how bright to set the ceiling light; a motion sensor can tell the actuator for the corridor lighting that someone has entered the corridor, or tell the room thermostat that there is no one in the room any more, so it can turn down the temperature.

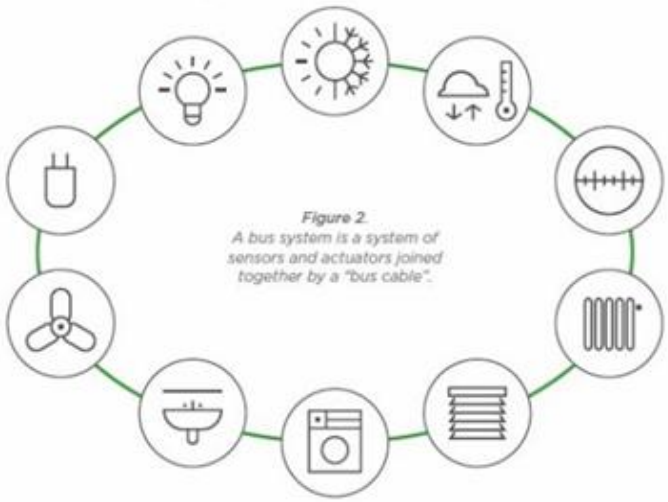


Figure 2
A bus system is a system of sensors and actuators joined together by a "bus cable".

Thanks to their decentralised structure, KNX bus systems can be modified and added to exactly as required. The smallest possible KNX application is a system linking two bus devices: a sensor and an actuator. This basic system can later be upgraded with as many devices as necessary to perform the desired control tasks. Theoretically a KNX system can consist of more than 50,000 devices. When extending a KNX system it is necessary to adhere to a specific topology.

Communication media:

Various communication media (and hence transmission methods) can be used for the exchange of data between devices in a KNX system:

1. KNX Twisted Pair (KNX TP)
   communication via a twisted pair data cable (bus cable)
2. KNX Powerline (KNX PL)
   uses the existing 230 V mains network
3. KNX Radio Frequency (KNX RF)
   communication via radio signal
4. KNX IP
   communication via Ethernet

Protocol:

The KNX system uses two Ethernet communication methods – tunneling and routing – both of which use the UDP protocol. Tunneling is used to access the bus from a local network or the internet for purposes of e. g. programming the KNX installation, while routing is used for exchanging telegrams over an Ethernet network, e. g. to couple two KNX TP systems via Ethernet.

The KNX protocols for these two communication methods are called KNXnet/IP routing and KNXnet/IP tunneling. IP communication in KNX can be explained using the OSI reference model (Fig. 15). Communication takes place via the application layer (which generates the KNXnet/IP telegram), the transport layer (UDP), the network layer (IP), and Ethernet – the physical layer. Like with the TP protocol, additional information for the respective layer (the header) is always added to the KNXnet/IP information.

Figure 16. KNXnet/IP telegram

Telegram structure
The KNXnet/IP telegram contains some further information in addition to that in the KNX TP telegram (Fig. 16):
Header Length: The header length is always the same. This information is still sent, however, because the header length may change in a later version of the protocol. The purpose of the header is to identify the start of the telegram.
Protocol Version: This indicates what version of the KNXnet/IP protocol applies.
KNXnet/IP Service Type Identifier: The KNXnet/IP Service Type Identifier indicates the action that is to be carried out.
Total Length: This field indicates the total length of the KNXnet/IP telegram.
KNXnet/IP-Body: This field contains the payload.

Zigbee

2.9 Zigbee:

ZigBee and IEEE 802.15.4 are standards-based protocols that provide the network infrastructure required for wireless sensor network applications. 802.15.4 defines the physical and MAC layers, and ZigBee defines the network and application layers.

For sensor network applications, key design requirements revolve around long battery life, low cost, small footprint, and mesh networking to support communication between large numbers of devices in an interoperable and multi-application environment.

There are numerous applications that are ideal for the redundant, self-configuring and self-healing capabilities of ZigBee wireless mesh networks. Key ones include

- Energy Management and Efficiency—To provide greater information and control of energy usage, provide customers with better service and more choice, better manage resources, and help to reduce environmental impact.
- Home Automation—To provide more flexible management of lighting, heating and cooling, security, and home entertainment systems from anywhere in the home.
- Building Automation—To integrate and centralize management of lighting, heating, cooling and security.
- Industrial Automation—To extend existing manufacturing and process control systems reliability.

Motivation for ZigBee:

The ZigBee standard was developed to address the following needs:

1. Low cost
2. Secure
3. Reliable and self healing
4. Flexible and extendable
5. Low power consumption
6. Easy and inexpensive to deploy
7. Global with use of unlicensed radio bands
8. Integrated intelligence for network set-up and message routing

ZigBee is the only standards-based technology that addresses the unique needs of most remote monitoring and control sensory network applications.

About the ZigBee Alliance

The ZigBee Alliance is an association of over 285 companies working together to enable reliable, cost-effective, low-power, wirelessly networked, monitoring and control products based on an open global standard. Their focus is on the following:

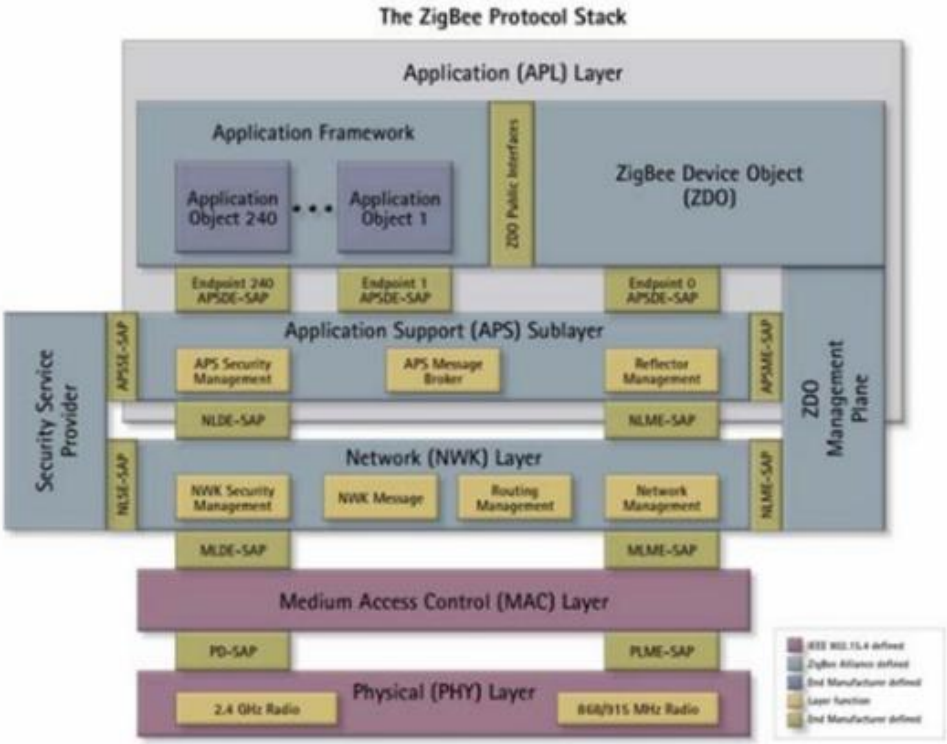
1. Defining the network, security and application software layers
2. Providing interoperability and conformance testing specifications
3. Promoting the ZigBee brand globally to build market awareness
4. Managing the evolution of the technology

The ZigBee Protocol Stack:

ZigBee sits on top of the IEEE 802.15.4 PHY and MAC layers. Each layer performs a specific set of services for the layer above. Each service entity provides an interface to the upper layer through a service access point (SAP).

Application (APL) Layer:

The top layer in the ZigBee protocol stack consists of the Application Framework, ZigBee Device Object (ZDO), and Application Support (APS) Sublayer.



Application Framework:

Provides a description of how to build a profile onto the ZigBee stack (to help ensure that profiles can be generated in a consistent manner). It also specifies a range of standard data types for profiles, descriptors to assist in service discovery, frame formats for transporting data, and a key value pair construct to rapidly develop simple attribute-based profiles.

Application Objects:

Software at an endpoint that controls the ZigBee device. A single ZigBee node supports up to 240 application objects. Each application object supports endpoints numbered between 1 and 240 (with endpoint 0 reserved for the ZigBee Device Object (ZDO)).

ZigBee Device Object (ZDO):

Defines the role of a device within the network (coordinator, router or end device), initiates and/or responds to binding and discovery requests, and establishes a secure relationship between network devices. It also provides a rich set of management commands defined in the ZigBee Device Profile (used in ZigBee commissioning). The ZDO is always endpoint zero.

ZDO Management Plane:

Facilitates communication between the APS and NWK layers with the ZDO. Allows the ZDO to deal with requests from applications for network access and security using ZDP (ZigBee Device Profile) messages.

Application Support (APS) Sublayer:

Responsible for providing a data service to the application and ZigBee device profiles. It also provides a management service to maintain binding links and the storage of the binding table itself.

Security Service Provider (SSP):

Provides security mechanisms for layers that use encryption (NWK and APS). Initialized and configured through the ZDO.

Network (NWK) Layer:

Handles network address and routing by invoking actions in the MAC layer. Its tasks include starting the network (coordinator), assigning network addresses, adding and removing network devices, routing messages, applying security, and implementing route discovery.

The ZigBee Network:

Device Types

ZigBee networks include the following device types:

1. Coordinators
2. Routers
3. End devices

Coordinator:

This device starts and controls the network. The coordinator stores information about the network, which includes acting as the Trust Center and being the repository for security keys.

Router:

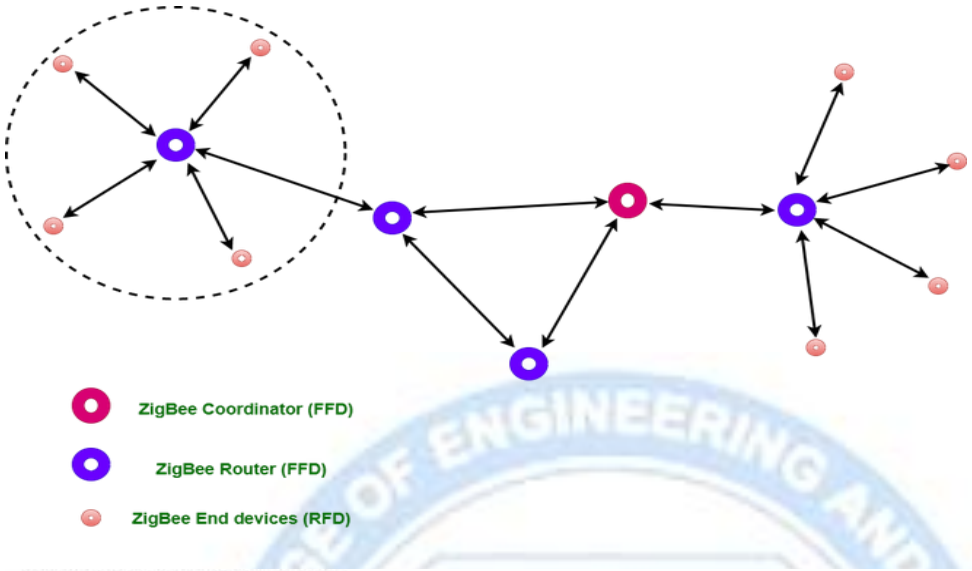
These devices extend network area coverage, dynamically route around obstacles, and provide backup routes in case of network congestion or device failure. They can connect to the coordinator and other routers, and also support child devices.

End Devices:

These devices can transmit or receive a message, but cannot perform any routing operations. They must be connected to either the coordinator or a router, and do not support child devices.

Benefits:

1. This topology is highly reliable and robust. Should any individual router become inaccessible, alternative routes can be discovered and used.
2. The use of intermediary devices in relaying data means that the range of the network can be significantly increased, making mesh networks highly scalable.
3. Weak signals and dead zones can be eliminated by simply adding more routers to the network.



Mesh Network Topology:

Mesh topology, also called peer-to-peer, consists of a mesh of interconnected routers and end devices. Each router is typically connected through at least two pathways, and can relay messages for its neighbors.

As shown in the image above, a mesh network contains a single coordinator, and multiple routers and end devices.

Mesh topology supports "multi-hop" communications, through which data is passed by hopping from device to device using the most reliable communication links and most cost-effective path until its destination is reached.

The multi-hop ability also helps to provide fault tolerance, in that if one device fails or experiences interference, the network can reroute itself using the remaining devices.

Network layer:

Network layer is also known as transmission layer. It acts like a bridge between perception layer and application layer. It carries and transmits the information collected from the physical objects through sensors. The medium for the transmission can be wireless or wire based. It also takes the responsibility for connecting the smart things, network devices and networks to each other. Therefore, it is highly sensitive to attacks from the side of attackers. It has prominent security issues regarding integrity and authentication of information that is being transported in the network.

Common security threats and problems to network layers are:

Denial of Service (DoS) Attack: A DoS attack is an attack to prevent authentic users from accessing devices or other network resources. It is typically accomplished by flooding the targeted devices or network resources with redundant requests in an order to make it impossible or difficult for some or all authentic users to use them .

Main-in-The-Middle (MiTM) Attack: MiTM attack is an attack where the attacker secretly intercepts and alters the communication between sender and receiver who believe they are directly communicating with each other. Since an attacker controls the communication, therefore he or she can change messages according to their needs. It causes a serious threat to online security because they give the attacker the facility to capture and manipulate information in real time.

Storage Attack: The information of users is stored on storage devices or the cloud. Both storage devices and cloud can be attacked by the attacker and user's information may be changed toin correct details. The replication of information associated with the access of other information by different types of people provides more chances for attacks.

Exploit Attack: An exploit is any immoral or illegal attack in a form of software, chunks of data ora sequence of commands. It takes advantage of security vulnerabilities in an application, system or hardware. It usually comes with the aim of gaining control of the system and steals information stored on a network

Application Layer:

Application layer defines all applications that use the IoT technology or in which IoT has deployed. The applications of IoT can be smart homes, smart cities, smart health, animal tracking, etc. It has the responsibility to provide the services to the applications. The services may be varying for each application because services depend on the information that is collected by sensors.

There are many issues in the application layer in which security is the key issue. In particular, when IoT is used in order to make a smart home, it introduces many threats and vulnerabilities from the inside and outside. To implement strong security in an IoT based smart home, one of the main issues is that the devices used in smart homes have weak computational power and a low amount of storage such as ZigBee. Common security threats and problem of application layer are:

Sensors

Cross Site Scripting: It is an injection attack. It enables an attacker to insert a client-side script, such as java script in a trusted site viewed other users. By doing so, an attacker can completely change the contents of the application according to his needs and use original information in an illegal way.

Malicious Code Attack: It is a code in any part of software intended to cause undesired effects and damage to the system. It is a type of threat that may not be blocked or controlled by the useof anti-virus tools. It can either activate itself or be like a program requiring a user's attention to perform an action.

The ability of dealing with Mass Data: Due to a large number of devices and a massive amount of data transmission between users, it has no ability to deal with data processing according to the requirements. As a result, it leads to network disturbance and data loss.