IoT Protocols

Chapter Three

1. Protocol Standardization for IoT

- For ensuring interoperability, scalability, and security in the Internet of Things (IoT) ecosystem
- define common rules, formats, and procedures for communication, data exchange, and interoperability between IoT devices, networks, and platforms

Key Protocols:

1. Internet Protocol Suite (TCP/IP)
    - network layer (IP), transport layer (TCP, UDP), and application layer (HTTP, MQTT, CoAP) communication
    - IPv6
2. MQTT (Message Queuing Telemetry Transport)
    - MQTT is a lightweight, publish-subscribe messaging protocol
    - designed for IoT applications with low bandwidth, high-latency, and unreliable networks
    - simplicity, efficiency, and support for asynchronous, event-driven messaging

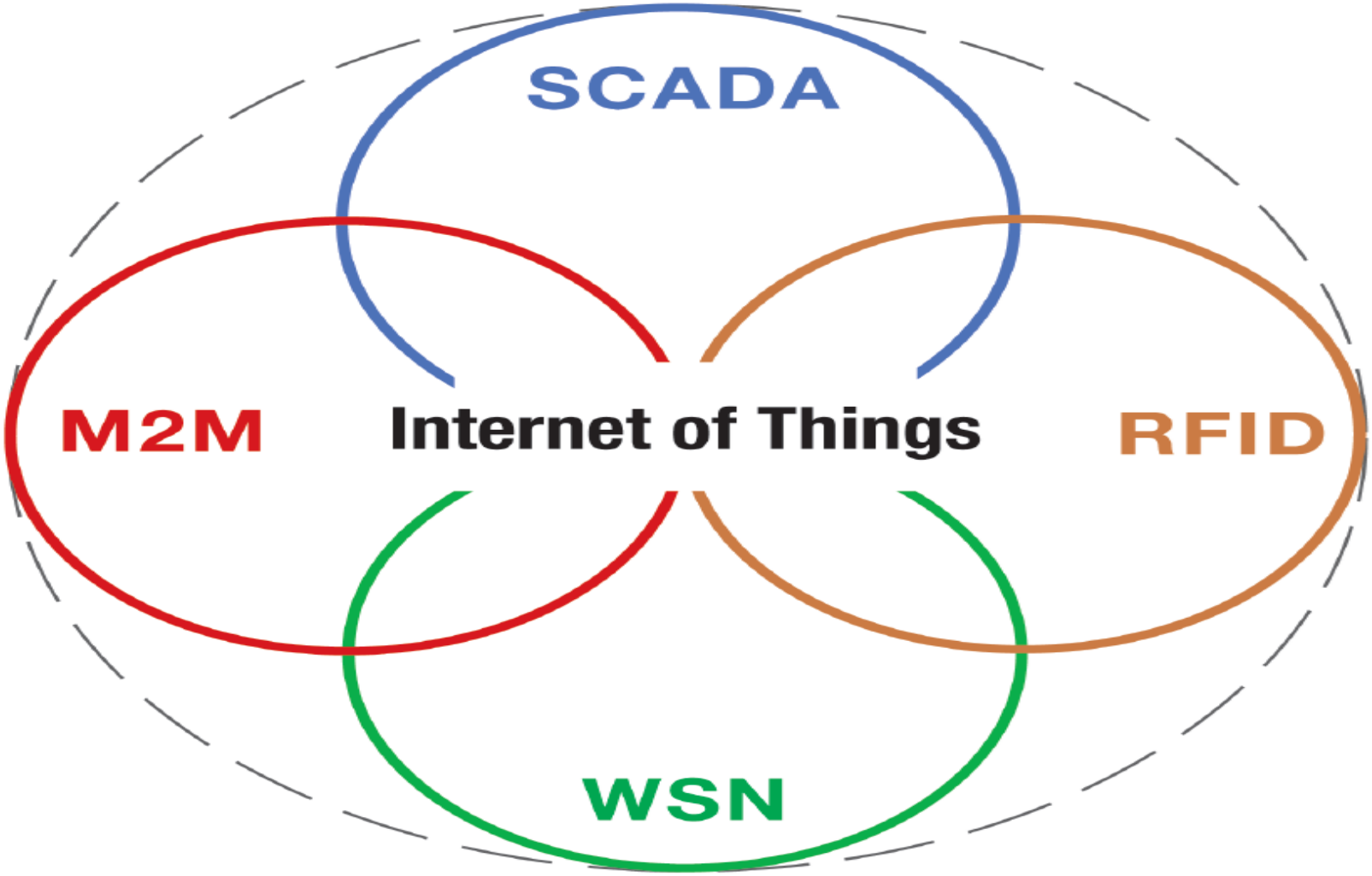
3. CoAP (Constrained Application Protocol)
    - CoAP is a lightweight, RESTful protocol
    - designed for resource-constrained IoT devices with limited memory, processing power, and energy resources
    - CoAP enables efficient communication between IoT devices and web services using standard HTTP methods (GET, POST, PUT, DELETE) over UDP or DTLS (Datagram Transport Layer Security)
4. AMQP (Advanced Message Queuing Protocol)
    - AMQP is an open standard messaging protocol
    - designed for reliable, interoperable messaging between applications, services, and devices
    - AMQP provides features such as message queuing, routing, reliability, and security
    - making it suitable for enterprise IoT applications with stringent requirements for messaging and integration
5. DDS (Data Distribution Service)
    - DDS is a standardized middleware protocol for real-time, scalable, and reliable data distribution and communication in distributed systems, including IoT
    - DDS provides publish-subscribe messaging, Quality of Service (QoS) policies, and data-centric communication models suitable for mission-critical IoT applications in industries such as aerospace, healthcare, and industrial automation

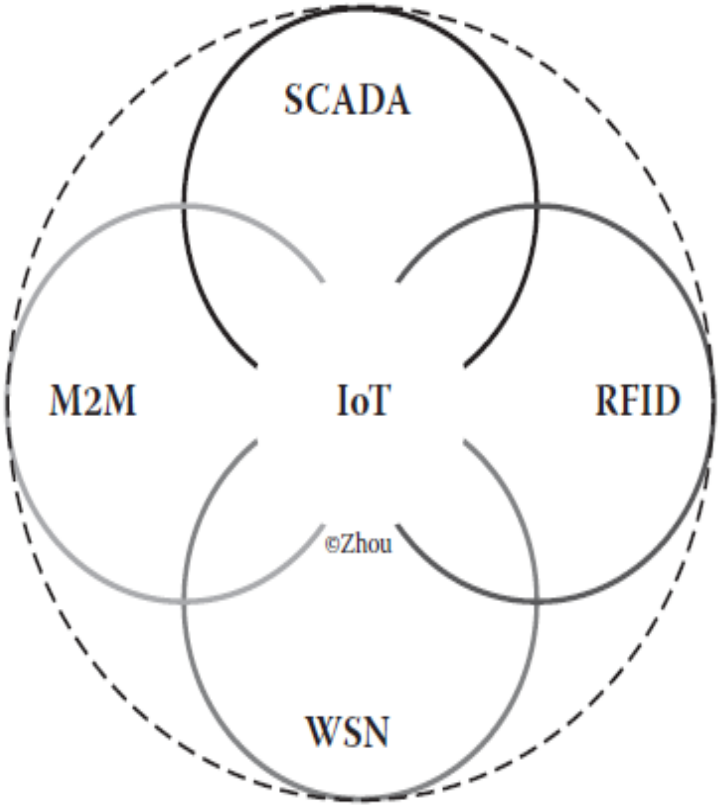
6. LoRaWAN (Long Range Wide Area Network)
    - LoRaWAN is a standardized communication protocol for low-power, wide-area networks (LPWANs)
    - designed for long-range communication and low-power IoT applications
    - LoRaWAN defines the physical layer (LoRa modulation) and MAC layer (LoRaWAN protocol) for secure, bidirectional communication between IoT devices and gateways
7. Thread
    - Thread is a standardized IPv6-based mesh networking protocol designed for smart home and building automation applications
    - Thread provides secure, reliable, and low-power communication between IoT devices, gateways, and cloud services, enabling interoperability and scalability in smart home ecosystems
8. OneM2M (One Machine-to-Machine)
    - OneM2M is a global standards initiative for M2M (Machine-to-Machine) and IoT interoperability
    - aiming to define common specifications and APIs for seamless communication and integration between IoT systems and platforms
    - OneM2M provides a common service layer abstraction and framework for IoT applications, enabling cross-domain interoperability and scalability

3.2 Efforts

- Efforts in the field of IoT (Internet of Things) encompass a wide range of activities, initiatives, and collaborations aimed at advancing the development, adoption, and deployment of IoT technologies and applications

1. **Standardization and Interoperability:**
    - Standardization organizations such as the IETF (Internet Engineering Task Force), IEEE (Institute of Electrical and Electronics Engineers), ITU (International Telecommunication Union), and industry consortia work on developing standards and protocols to ensure interoperability, security, and scalability in IoT ecosystems
    - Efforts focus on defining common protocols, data formats, communication models, and security mechanisms to enable seamless integration and communication between diverse IoT devices, networks, and platforms
2. Open Source Projects and Communities
3. Research and Development
4. Industry Consortia and Alliances
5. Regulatory and Policy Frameworks
6. Education and Awareness
7. Collaborative Projects and Initiatives


SCADA
M2M
Internet of Things
RFID
WSN



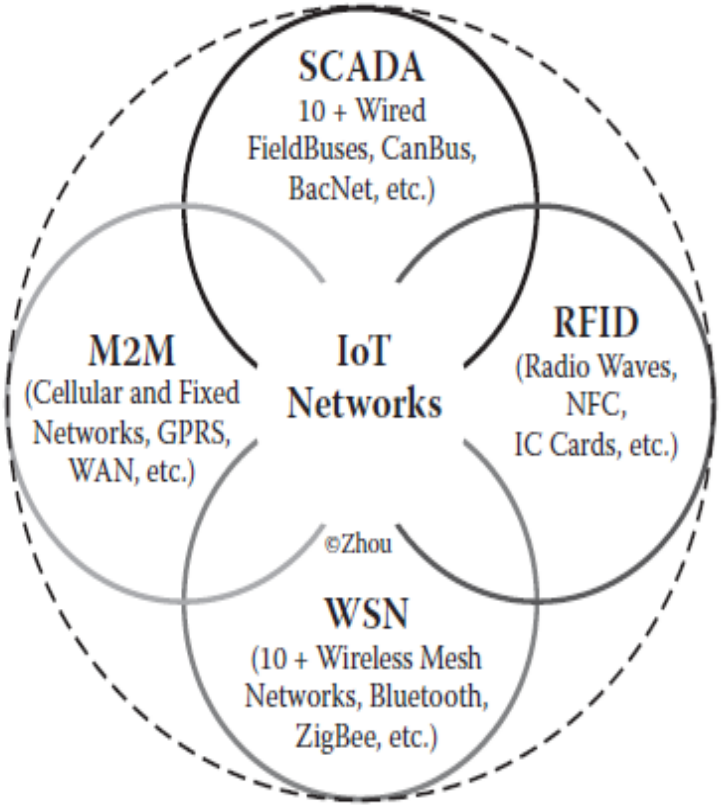


Figure The four pillars of IoT paradigms and related networks.

3.3 M2M and WSN Protocols

1. Machine-to-Machine (M2M) Protocols:
    - MQTT (Message Queuing Telemetry Transport)
    - CoAP (Constrained Application Protocol)
    - AMQP (Advanced Message Queuing Protocol)
    - DDS (Data Distribution Service)
2. Wireless Sensor Networks (WSN) Protocols
    - IEEE 802.15.4
    - 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks)
    - RPL (Routing Protocol for Low-Power and Lossy Networks)
    - Zigbee

3.4 SCADA and RFID Protocols

1. SCADA Protocols
    - Modbus
    - DNP3 (Distributed Network Protocol)
    - OPC UA (Open Platform Communications Unified Architecture)
    - IEC 61850 (International Electrotechnical Commission Standard 61850)
2. RFID Protocols
    - EPC Gen2 (Electronic Product Code Generation 2)
    - ISO/IEC 14443 (NFC)
    - ISO/IEC 15693
    - LLRP (Low-Level Reader Protocol)

3.5 Unified data standards and protocols

1. JSON (JavaScript Object Notation)
2. XML (Extensible Markup Language)
3. Message Queuing Telemetry Transport (MQTT)
4. CoAP (Constrained Application Protocol)
5. OPC UA (Open Platform Communications Unified Architecture)
6. JSON-LD (JSON for Linked Data)
7. Schema.org

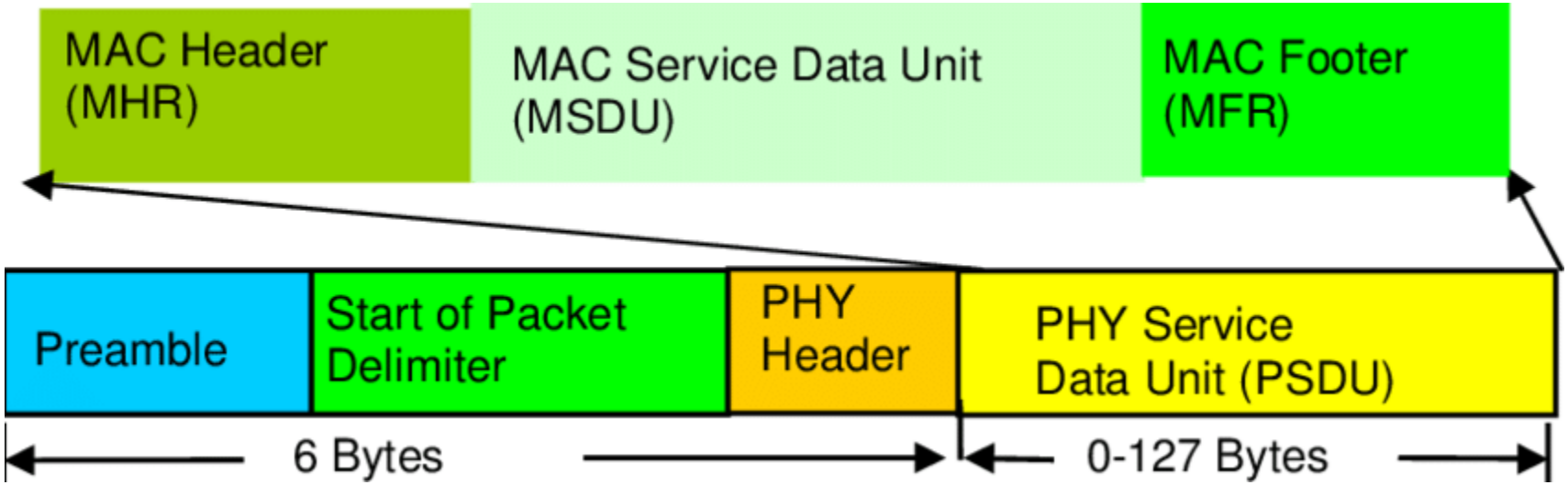
3.6 IEEE 802.15.4

- IEEE 802.15.4 is a standard for low-power, low-data-rate wireless communication in personal area networks (PANs)
- It defines the physical (PHY) and media access control (MAC) layers for short-range, low-rate wireless connectivity, making it suitable for applications with constrained power sources, limited bandwidth, and low-complexity devices
- Applications such as home automation, industrial monitoring, healthcare, smart agriculture, and asset tracking etc.
- It provides a cost-effective, energy-efficient solution for short-range communication between devices operating in the same PAN

Physical Layer (PHY)

- The PHY layer of IEEE 802.15.4 specifies the radio transmission characteristics, modulation schemes, and frequency bands for wireless communication
- It supports multiple frequency bands, including 2.4 GHz, 868 MHz, and 915 MHz, with varying regional regulations and propagation characteristics
- IEEE 802.15.4 PHY defines various data rates (e.g., 250 kbps, 100 kbps) and modulation schemes (e.g., O-QPSK, BPSK) to accommodate different application requirements and environmental conditions

3. **Media Access Control (MAC) Layer:**
    - The MAC layer of IEEE 802.15.4 defines the rules and procedures for accessing the shared wireless medium and managing communication between devices.
    - It supports different network topologies, including star, peer-to-peer, and mesh networks, allowing flexible deployment and scalability in PAN applications.
    - IEEE 802.15.4 MAC provides mechanisms for channel access, synchronization, frame formatting, error detection, and energy management to optimize performance and reliability in low-power wireless networks.
4. **Frame Structure:**
    - IEEE 802.15.4 frames consist of a preamble, header, payload, and frame check sequence (FCS), following a predefined structure for data transmission and reception.
    - Frames can be transmitted in beacon-enabled or non-beacon-enabled modes, depending on the network topology and synchronization requirements.
    - Different frame types (e.g., data, acknowledgment, beacon) and addressing modes (e.g., short address, extended address) are supported to accommodate various communication scenarios and message formats.

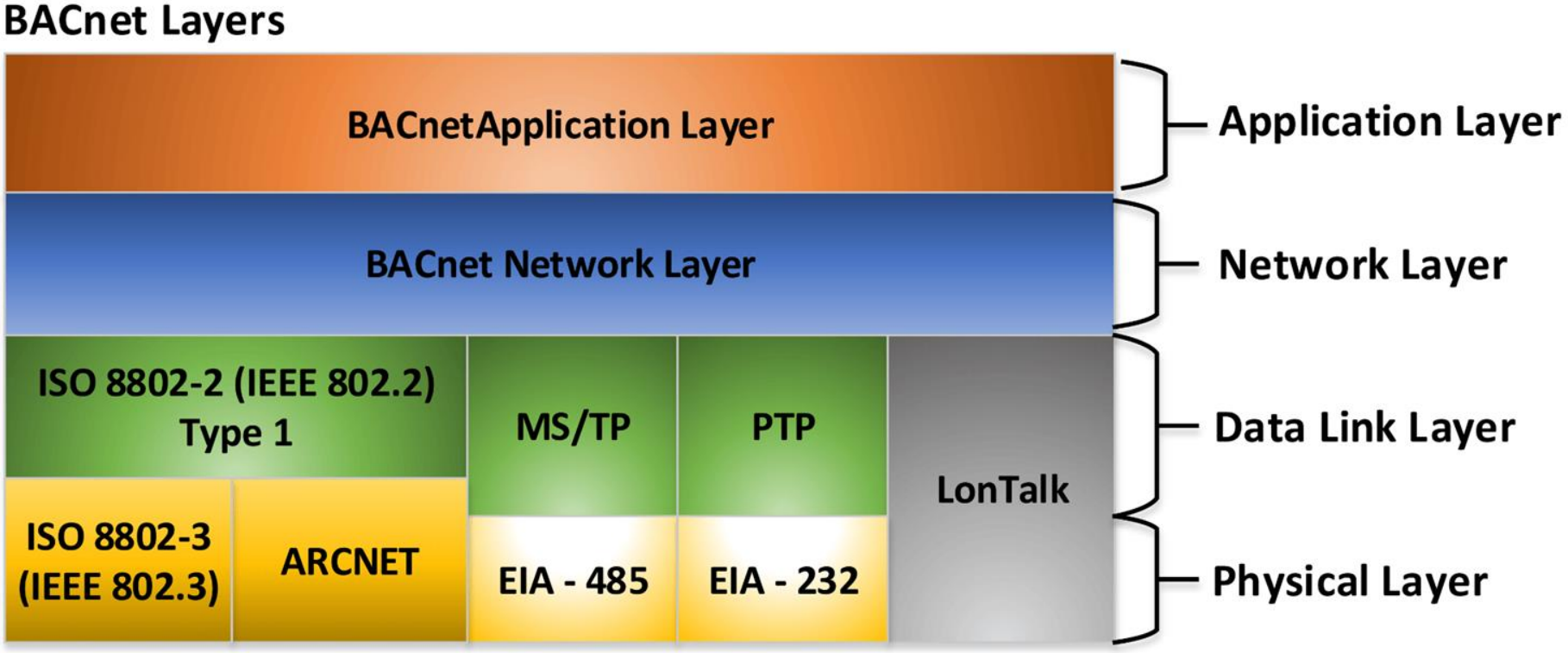

MAC Header
(MHR)
MAC Service Data Unit
(MSDU)
MAC Footer
(MFR)
Preamble
Start of Packet
Delimiter
PHY
Header
PHY Service
Data Unit (PSDU)
6 Bytes
0-127 Bytes

5. **Security Features:**
    - IEEE 802.15.4 includes provisions for securing communication and protecting against unauthorized access, interception, and tampering.
    - It supports encryption, authentication, and key management mechanisms to ensure confidentiality, integrity, and authenticity of data exchanged between devices.
    - Security features include symmetric and asymmetric encryption algorithms, message authentication codes (MACs), and secure key exchange protocols for establishing secure communication channels in IEEE 802.15.4 networks
6. Interoperability:
    - IEEE 802.15.4 is designed to promote interoperability and compatibility between devices from different manufacturers and vendors.
    - It defines a common standard for PHY and MAC layers, enabling seamless integration and communication between IEEE 802.15.4-compliant devices and systems.
    - IEEE 802.15.4-based devices can operate in mixed networks with other wireless technologies (e.g., Wi-Fi, Bluetooth) and coexist in the same environment without significant interference or compatibility issues

3.7 BACNet (Building Automation and Control Network) Protocol

- BACnet (Building Automation and Control Networks) is a communication protocol specifically designed for building automation and control systems
- It is an ANSI/ASHRAE (American National Standards Institute/American Society of Heating, Refrigerating and Air-Conditioning Engineers) standard that facilitates communication between building automation devices and systems from different manufacturers
- BACnet is widely used in commercial and industrial buildings for integrating and controlling various building systems, including HVAC (Heating, Ventilation, and Air Conditioning), lighting, access control, fire detection, and security systems
- It enables interoperability and data exchange between different building automation devices, such as sensors, actuators, controllers, and management software, regardless of vendor or protocol

Communication Architecture



- BACnet uses a client-server communication model with standardized objects, services, and protocols for device interaction.
- BACnet devices can act as clients, servers, or both, allowing for flexible configurations and system architectures.
- Communication between BACnet devices is based on a request-response mechanism, where clients send requests (e.g., read property, write property) to servers, and servers respond with the requested data or perform the requested action

Object-Oriented Model:

- BACnet defines a set of standardized objects and properties for representing building automation entities and data points.
- Objects include analog inputs/outputs, binary inputs/outputs, multi-state inputs/outputs, schedules, trends, alarms, and more, each with predefined properties for configuration and monitoring.
- The object-oriented model provides a common framework for device configuration, data representation, and interoperability in BACnet-based systems.

Services and Protocols:

- BACnet defines a set of standard services and protocols for communication between devices, including:
  - Object Access Services: Read property, write property, read property multiple, write property multiple, subscribe COV (Change of Value).
  - Alarm and Event Services: Acknowledge alarm, get alarm summary, acknowledge multiple alarms, get event information.
  - File Access Services: Read file, write file, delete file, get file information.
- BACnet messages can be transmitted over various network protocols, including Ethernet, ARCNET, BACnet/IP (Internet Protocol), BACnet MSTP (Master-Slave/Token-Passing), BACnet over LonTalk, and BACnet over Zigbee. (Local Operating Network-Lon)

Interoperability:

- Interoperability is a key feature of BACnet, enabling seamless integration and communication between devices from different manufacturers and vendors.
- BACnet-compliant devices must adhere to the standard's requirements for object types, services, encoding rules, and network protocols to ensure compatibility and interoperability.
- BACnet testing and certification programs help validate device compliance and ensure interoperability in real-world deployments

Security:

- BACnet includes provisions for securing communication and protecting against unauthorized access, tampering, and data breaches.
- Security features may include authentication, encryption, access control lists (ACLs), and secure communication protocols (e.g., BACnet/IP with TLS/SSL).
- Implementing secure BACnet networks helps safeguard sensitive building data and mitigate cybersecurity risks in building automation systems

8. Modbus

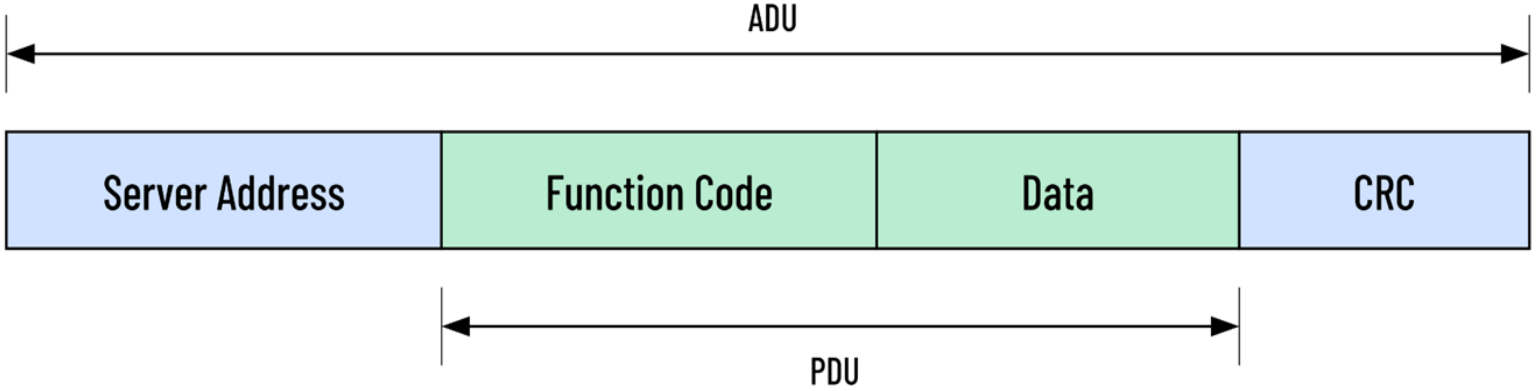
- Modbus is a widely used communication protocol in the field of industrial automation and control systems
- It was developed in the late 1970s by Modicon (now Schneider Electric) for use with their programmable logic controllers (PLCs)
- Modbus is primarily used for communication between PLCs, RTUs (Remote Terminal Units), SCADA systems, HMIs (Human-Machine Interfaces), and other industrial devices.
- It facilitates the exchange of process data, control commands, and status information between devices in industrial automation and control applications.

Protocol Variants:

- Modbus has several variants, including:
  - Modbus RTU (Remote Terminal Unit): This variant uses a compact binary representation of data and is typically transmitted over serial communication links (RS-232 or RS-485).
  - Modbus ASCII: Similar to Modbus RTU, but data is encoded in ASCII characters, making it more human-readable for troubleshooting and diagnostics.
  - Modbus TCP/IP: This variant encapsulates Modbus messages within TCP/IP packets and is used for communication over Ethernet networks, enabling integration with IT infrastructure and internet connectivity.

Message Format:

- Modbus messages consist of a header followed by data fields, with each field containing information such as function codes, addresses, data values, and error checking.
- The most commonly used functions in Modbus messages include:
  - Read Holding Registers
  - Read Input Registers
  - Write Single Register
  - Write Multiple Registers
  - Read Coils
  - Write Single Coil
  - Write Multiple Coils
- These functions allow for reading and writing data to various types of memory registers within Modbus-enabled devices



Modbus Message Frame Format

Master-Slave Architecture:

- Modbus communication follows a master-slave architecture, where one device (the master) initiates communication and controls the data exchange process, while one or more devices (slaves) respond to requests from the master.
- The master sends requests to read or write data to specific registers in the slave devices, and slaves respond with the requested data or acknowledge the command.

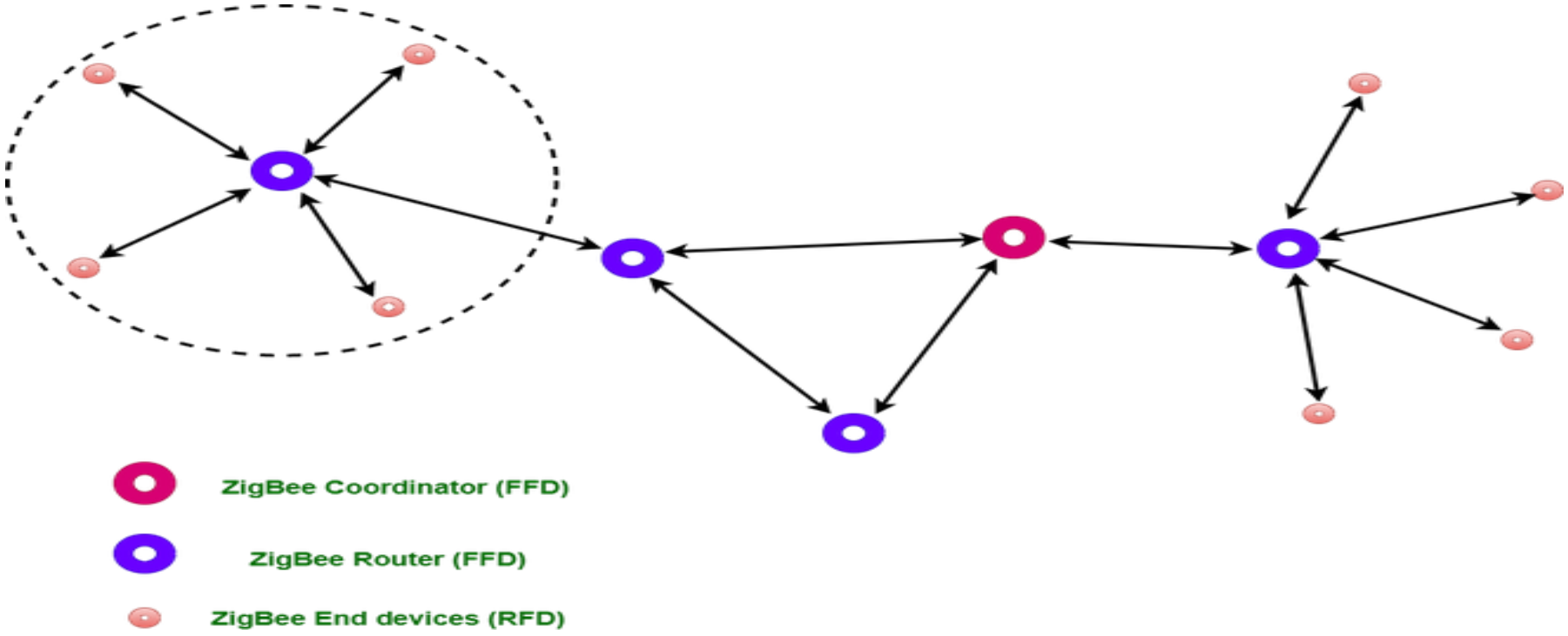
Interoperability:

- Modbus is known for its simplicity and ease of implementation, which has contributed to its widespread adoption in the industrial automation industry.
- It is a de facto standard supported by many industrial equipment vendors, making it easy to integrate Modbus-compatible devices from different manufacturers into a single control system.
- Interoperability is further facilitated by the availability of Modbus gateways and converters, which allow Modbus devices to communicate with other industrial protocols such as Profibus, DeviceNet, and Ethernet/IP

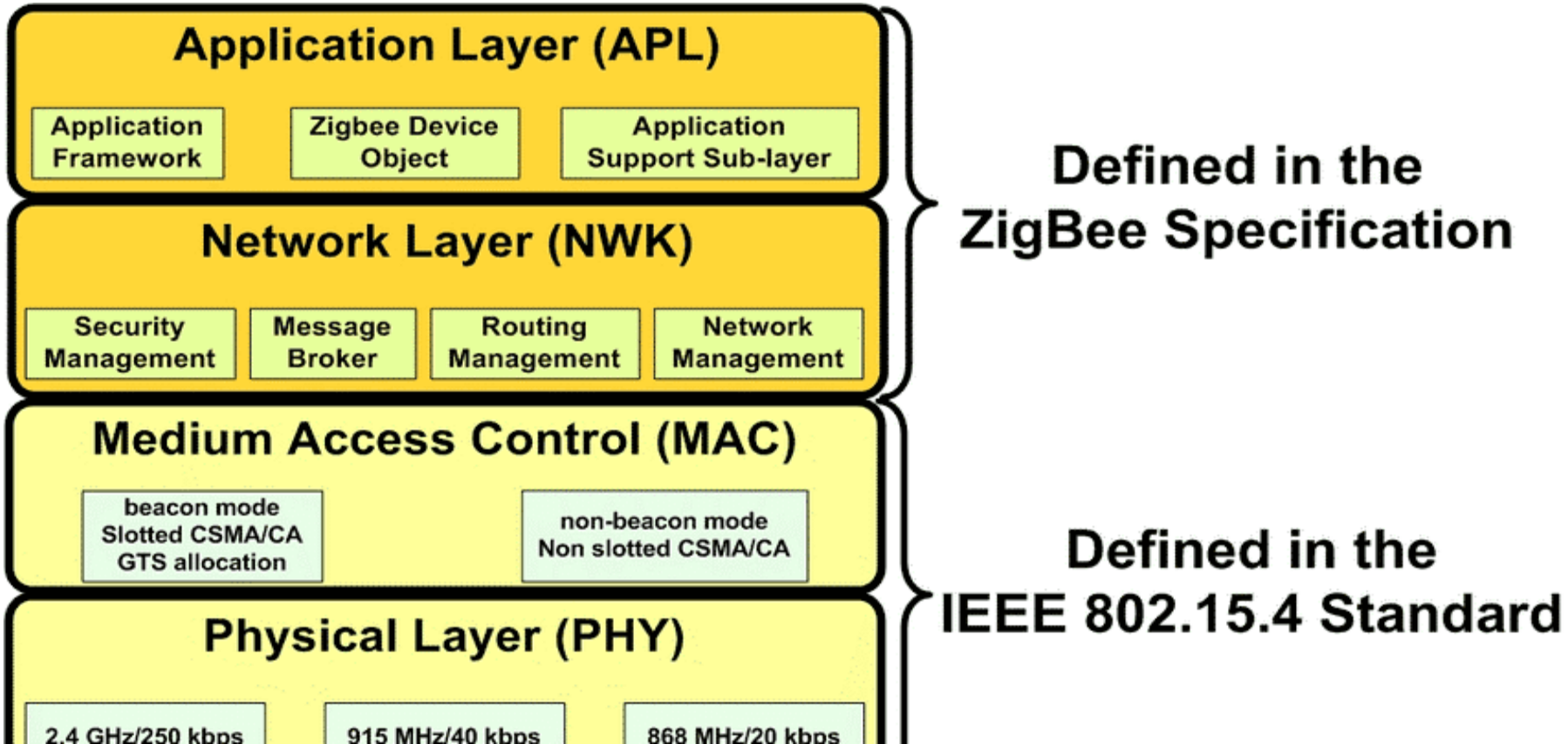
3.9 Zigbee Architecture

- Zigbee is a low-power, low-data-rate wireless communication protocol designed for short-range, low-cost, and low-complexity applications such as home automation, industrial control, and sensor networks
- The Zigbee architecture consists of several key components and layers, each serving specific functions in the communication process
- Here's an overview of the Zigbee architecture

Zigbee System Architecture



Zigbee Protocol Architecture



Zigbee framework structure

1. Zigbee Devices:
    - Zigbee devices are the endpoints in a Zigbee network and can take various forms, including sensors, actuators, controllers, and routers.
    - Devices communicate with each other using Zigbee's standardized protocol stack and can form self-organizing, multi-hop mesh networks to extend the coverage and reliability of the network.
2. Zigbee Coordinator:
    - The Zigbee coordinator is a central node in a Zigbee network responsible for initiating and managing network formation, device association, and network security.
    - It typically serves as the network controller and may also provide gateway functionality to connect Zigbee networks with other networks, such as Wi-Fi or Ethernet.
3. Zigbee Network (Router):
    - A Zigbee network consists of multiple Zigbee devices organized into a mesh topology, where devices communicate with each other directly or through intermediate routers.
    - Devices in the network form dynamic connections and can join or leave the network dynamically, allowing for flexible and scalable deployments.

Zigbee Protocol Stack

- The Zigbee protocol stack consists of several layers, each responsible for specific functions in the communication process:
  - Application Layer: Defines the application-specific functionality and profiles for Zigbee devices, such as home automation, lighting control, or smart energy.
  - Network Layer: Handles network formation, device addressing, routing, and data forwarding in the Zigbee network.
  - MAC (Media Access Control) Layer: Manages access to the wireless medium, frame formatting, and channel coordination to ensure efficient and reliable communication.
  - PHY (Physical) Layer: Specifies the radio transmission characteristics, modulation schemes, and frequency bands used for wireless communication

3.10 Network Layer

- The network layer is responsible for several key functions that enable efficient and reliable communication among Zigbee devices

Key Functions of the Zigbee Network Layer

- Network Formation and Management:
  - Network Initialization: Facilitates the creation of a new Zigbee network by a coordinator. The coordinator selects a suitable radio channel and PAN ID to avoid interference with other networks.
  - Address Assignment: Manages the assignment of unique 16-bit short addresses to each device in the network. The coordinator assigns addresses to routers and end devices.
- Routing:
  - Tree Routing: Utilizes a hierarchical tree structure for routing. Each device (except end devices) maintains information about its children and parent. This method is simple but can be inefficient in terms of path length.
  - Mesh Routing (AODV): Implements the Ad hoc On-Demand Distance Vector (AODV) routing protocol for more flexible and efficient routing. This allows for dynamic route discovery and maintenance, enabling devices to find the shortest path to the destination

3. **Network Topology:**
    - Star Topology: Centralized control where the coordinator communicates directly with all devices.
    - Tree Topology: Devices form a hierarchical structure with the coordinator at the root.
    - Mesh Topology: Devices can communicate directly or indirectly through multiple hops, providing redundancy and resilience
4. **Joining and Leaving Networks:**
    - Device Joining: Allows devices to join an existing network. The network layer handles the discovery of available networks and the joining process.
    - Device Leaving: Manages the dissociation of devices from the network, ensuring network stability and security
5. **Security:**
    - Authentication and Encryption: Ensures secure communication through encryption and authentication mechanisms. The network layer uses AES-128 encryption to protect data and control messages.
    - Key Management: Manages network security keys, including their distribution and refreshment

6. **Network Maintenance:**
    - Route Discovery and Repair: Continuously monitors the network for link failures and dynamically repairs routes as needed.
    - Neighbor Table Management: Maintains information about neighboring devices, which is essential for efficient routing and network maintenance.
7. **Data Handling:**
    - Frame Formatting: Handles the encapsulation and decapsulation of network frames. The network layer formats data packets with necessary headers and trailers.
    - Data Delivery: Ensures reliable delivery of data frames to the appropriate destination, handling retransmissions if necessary

3.11 6LowPAN

- 6LoWPAN enables the integration of low-power wireless devices into IPv6 networks, allowing them to communicate with other IPv6-enabled devices and services on the Internet
- It defines mechanisms for compressing IPv6 headers and adapting IPv6 packets to fit within the constraints of low-power wireless networks, such as IEEE 802.15.4.

Header Compression:

- One of the key features of 6LoWPAN is header compression, which reduces the overhead associated with IPv6 packet headers to conserve bandwidth and minimize energy consumption.
- Header compression techniques include stateless compression, context-based compression, and link-layer adaptation, which optimize the representation of IPv6 headers for transmission over low-power wireless links

Fragmentation and Reassembly:

- 6LoWPAN supports fragmentation and reassembly of IPv6 packets to accommodate the smaller maximum transmission unit (MTU) sizes and variable data rates of IEEE 802.15.4 networks.
- Large IPv6 packets are divided into smaller fragments for transmission over the network, and the receiving devices reassemble the fragments into complete IPv6 packets

Neighbor Discovery:

- Neighbor Discovery in 6LoWPAN enables devices to discover and maintain information about neighboring devices on the network, including their IPv6 addresses and link-layer identifiers.
- Neighbor Discovery protocols facilitate address resolution, link-layer adaptation, and route discovery in 6LoWPAN networks, enabling efficient communication between devices

Routing:

- Routing protocols such as RPL (Routing Protocol for Low-Power and Lossy Networks) are commonly used in 6LoWPAN networks to establish and maintain routes between devices and gateways.
- RPL is specifically designed for low-power and lossy networks, providing efficient route discovery, optimization, and management for IPv6 communication in 6LoWPAN deployments.

Security:

- Security is an important consideration in 6LoWPAN deployments to protect data transmission and ensure the integrity and confidentiality of communication between devices.
- Security mechanisms may include encryption, authentication, and key management protocols to secure IPv6 packets and prevent unauthorized access or tampering.

3.12 CoAP

- CoAP (Constrained Application Protocol) is a specialized web transfer protocol designed for constrained devices and low-power, low-bandwidth networks such as those found in IoT (Internet of Things) deployments.
- CoAP is specifically optimized for use with constrained nodes and networks, offering lightweight communication for resource-constrained devices

1. **RESTful Protocol:**
    - CoAP follows the principles of Representational State Transfer (REST) architecture, providing a lightweight alternative to HTTP for constrained environments.
    - It defines methods such as GET, POST, PUT, and DELETE for resource manipulation, making it easy to develop RESTful IoT applications using CoAP.

UDP-Based:

- CoAP is based on the User Datagram Protocol (UDP), which is lightweight and connectionless, making it suitable for constrained devices and lossy networks.
- UDP provides simplicity and low overhead compared to TCP, enabling efficient communication with minimal protocol overhead.

Resource-Oriented Communication:

- CoAP is designed around the concept of resources, which represent information or services exposed by IoT devices.
- Resources are identified by URIs (Uniform Resource Identifiers) and can be accessed and manipulated using CoAP methods such as GET, POST, PUT, and DELETE.

Lightweight Header Format:

- CoAP uses a compact header format to minimize message size and conserve bandwidth.
- The header includes fields for message type, message code, message ID, token, options, and payload length, providing essential information for message processing and routing.

URI and Content-Format Options:

- CoAP supports URI options for identifying resources and specifying resource hierarchy within a server.
- It also supports content-format options for indicating the media type of payload data, allowing devices to exchange data in various formats such as text/plain, application/json, and application/xml

Observing Resources:

- CoAP includes support for observing resources, allowing clients to subscribe to resource updates and receive notifications when the resource state changes.
- Observing resources enable efficient event-driven communication and real-time data monitoring in IoT applications.

Security:

- CoAP supports Datagram Transport Layer Security (DTLS) for securing communication between devices.
- DTLS provides encryption, authentication, and integrity protection for CoAP messages, ensuring secure data exchange in IoT deployments