Mapping IoT Communications to the OSI Model

* How does IoT communications map to the Physical/Data Link Layer of the OSI Model?
  + At the Physical and Data Link layers, IoT communications involve the actual transmission of data over a physical device.
  + At Physical Level – this layer is concerning on how each physical device is connecting (physically) to the network with hardware – e.g: cable, wire, radio, wireless network.
  + At Data Level - this layer is responsible for framing data, physical addressing (MAC addresses), error detection, and flow control within a local network.
    - MAC protocols (e.g., CSMA/CA for Wi-Fi, TDMA/CSMA for Zigbee): Manage access to the shared physical medium.
    - Error detection and correction mechanisms: Ensure data integrity.
    - Framing: Structuring data into frames for transmission.
* How does IoT communications map to the Network/Transport Link Layer of the OSI Model?
  + These two layers are responsible for enabling end-to-end communication across different networks and ensuring reliable data delivery.
    - Network Layer: This layer is responsible for logical addressing (IP addresses), routing data packets across different networks, and determining the best path for data.
      * IPv6/IPv4: IPv6 is becoming increasingly important for IoT due to the massive number of devices requiring unique addresses.
      * IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN): A key adaptation layer that allows IPv6 packets to be sent over low-power wireless networks like IEEE 802.15.4, making it possible for resource-constrained IoT devices to participate in IP-based networks.
      * Routing Protocols:
        + RPL (Routing Protocol for Low-Power and Lossy Networks): A routing protocol specifically designed for resource-constrained IoT devices and mesh networks, often used with 6LoWPAN.
    - Transport Layer: This layer provides end-to-end communication between applications, ensuring reliable data transfer and flow control.
      * UDP (User Datagram Protocol): Often preferred in IoT for its low overhead, making it suitable for devices with limited resources and applications where occasional data loss is acceptable
      * TCP (Transmission Control Protocol): Used when reliable, ordered, and error-checked data delivery is critical (e.g., firmware updates, critical control commands).
      * DCCP (Datagram Congestion Control Protocol): Less common in general IoT but can be used for connection-oriented, unreliable datagram flow with congestion control.
* How does IoT communications map to the Session/Presentation/Application Layer of the OSI Model?
  + These three layers are dealing with inter-application communication, data formatting, and providing services directly to the end-user or other applications.
    - Session Layer – establishes, manages, and terminates communications sessions between applications.
    - Presentation Layer – responsible for data formatting, encryption, decryption, and compression, ensuring that data is presented in a readable format for the application layer.
    - Application Layer – directly interacts with software applications. It provides high-level services and protocols for specific IoT use cases

Low Power Communications

* Why do we need low power communications for IoT?
  + Extended Battery Life:
    - Most IoT devices run on batteries.
    - Low-power communication allows them to operate for months or even years without needing battery changes → cutting down on maintenance costs and effort, especially for widespread deployments.
  + Scalability:
    - To manage billions of connected devices, each one must be energy efficient.
    - Powering and maintaining a massive network of power-hungry devices would be logistically impossible and financially unsustainable.
  + Deployment Flexibility:
    - Low-power devices can be placed almost anywhere, including remote or hard-to-access locations without traditional power sources, broadening the scope of IoT applications.
  + Cost-Effectiveness:
    - Less power consumption means simpler, cheaper components and smaller batteries, which lowers the overall manufacturing cost of devices and reduces long-term operational expenses.
  + Miniaturization:
    - Lower power requirements mean less heat generation and smaller power components, enabling more compact and integrated device designs suitable for small products or wearables.
* What are the characteristics of LoWPAN technologies?
  + Concept: Localized, short-range, low-power networks, often for personal or small-area use. Think of devices talking within a room or building.
  + Range: Short (tens of meters). Ideal for personal area networks (PANs) or small-scale local deployments.
  + Power Consumption: Low. Designed for battery life from months to a few years.
  + Data Rates: Low. Suitable for small, infrequent data packets (e.g., sensor readings, simple commands). Not for streaming.
  + Networking: Often supports mesh networking, where devices can relay messages for neighbors, extending local reach and improving resilience.
  + IP Integration: Crucially, often supports IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN), allowing resource-constrained devices to participate directly in IP networks.
  + Cost: Generally low cost per node and relatively simple infrastructure.
  + Examples: Zigbee, Bluetooth Low Energy (BLE), Thread.
* What are the characteristics of LPWAN technologies?
  + Concept: Localized, short-range, low-power networks, often for personal or small-area use. Think of devices talking within a room or building.
  + Range: Short (tens of meters). Ideal for personal area networks (PANs) or small-scale local deployments.
  + Power Consumption: Low. Designed for battery life from months to a few years.
  + Data Rates: Low. Suitable for small, infrequent data packets (e.g., sensor readings, simple commands). Not for streaming.
  + Networking: Often supports mesh networking, where devices can relay messages for neighbors, extending local reach and improving resilience.
  + IP Integration: Crucially, often supports IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN), allowing resource-constrained devices to participate directly in IP networks.
  + Cost: Generally low cost per node and relatively simple infrastructure.
  + Examples: Zigbee, Bluetooth Low Energy (BLE), Thread.

Application Layer Communications

* What are the characteristics of RESTful communication?
  + REST, fundamentally an architectural style often implemented over HTTP/HTTPS, offers a familiar paradigm for web developers.
    - Statelessness:
      * Each client request is self-contained.
      * The server doesn't retain session information between requests.
      * While excellent for scalability (any server can handle any request), it can add overhead for repetitive interactions, which might be a concern for resource-constrained IoT devices or very frequent data points.
    - Client-Server Decoupling:
      * A clear separation, allowing independent evolution of the client (device) and server (cloud platform).
      * This is beneficial for large, distributed systems.
    - Cache-ability:
      * Responses can be marked as cacheable, reducing network traffic and server load for frequently accessed, unchanging data.
      * This can save bandwidth for devices if used correctly.
    - Uniform Interface:
      * Relies on standard HTTP methods (GET, POST, PUT, DELETE) to interact with identified resources (via URIs).
      * This consistency aids development but might be too verbose for extremely constrained devices.
    - Synchronous by Nature:
      * Typically, a client sends a request and waits for a response.
      * This can introduce latency and isn't ideal for event-driven, push-style communication common in sensor networks.
* What are the characteristics of MQTT communication?
  + MQTT is purpose-built for constrained environments and messaging patterns common in IoT.
    - Publish/Subscribe Model:
      * Clients publish messages to abstract "topics," and other clients subscribe to those topics.
      * A central broker handles message routing.
      * This inherently decouples senders from receivers, improving scalability and flexibility.
    - Lightweight and Efficient:
      * Designed with minimal overhead in mind, featuring compact message headers and a small code footprint.
      * This makes it highly efficient for constrained devices and low-bandwidth networks.
    - Asynchronous Communication:
      * Publishers and subscribers don't need direct knowledge of each other or simultaneous online presence.
      * The broker queues for messages.
      * This is crucial for devices with intermittent connectivity (e.g., LPWAN devices that "wake up, send, and sleep").
    - Quality of Service (QoS):
      * Offers distinct levels (0, 1, 2) for message delivery guarantees, allowing developers to balance reliability with overhead based on data criticality.
    - Persistent Sessions & Last Will:
      * Clients can maintain session state with the broker, ensuring messages are delivered even after a temporary disconnection.
      * The "Last Will and Testament" feature provides critical device status notification upon unexpected disconnections.
* What do we need from an application layer communications protocol to build a scalable IoT application?
* Efficiency for Constrained Devices: Protocols must be extremely lightweight in terms of payload size, header overhead, and computational demands (CPU, RAM). This directly impacts battery life and hardware cost.
* Network Resilience: Must perform well over unreliable, low-bandwidth, and high-latency networks (e.g., LPWANs). Features like asynchronous operation, message queuing, and QoS are critical.
* Massive Scalability: The architecture needs to gracefully handle millions to billions of concurrently connected devices and trillions of messages. This often points towards decoupled, broker-centric models (like pub/sub).
* Security Integration: Robust security mechanisms (encryption, authentication, authorization) are non-negotiable for data integrity and device security. This includes integration with transport layer security (TLS/DTLS).
* Bidirectional Communication: While many IoT devices are telemetry producers, the ability to receive commands, firmware updates, or configuration changes from the cloud is essential for management and control.
* Interoperability: Adherence to open standards is vital to prevent vendor lock-in and enable heterogeneous device ecosystems.
* Support for Diverse Data Models: Flexibility to handle various data formats (e.g., JSON, CBOR, Protobuf) efficiently, allowing developers to choose the most suitable one for their payload size and parsing needs.
* Device Management Capabilities: While not always solely at the application layer, protocols that facilitate remote provisioning, monitoring, and updating of devices (e.g., LwM2M often layered on CoAP) are highly advantageous for scalability.