Networking & Connectivity

1. Network Architecture

- IoT follows a **layered model** similar to TCP/IP:
    - **Perception Layer**: sensors & actuators (data collection).
    - **Network Layer**: communication (Wi-Fi, ZigBee, LoRa, LTE).
    - **Application Layer**: services & user apps (cloud dashboards, mobile apps).
- Devices usually talk **Device → Gateway → Cloud**.
- Purpose: ensures scalability and modular design.

2. Data Representation Formats

- **JSON (JavaScript Object Notation)**
    - Human-readable and widely used in REST APIs.
    - Example:
    - { "temperature": 26.4, "humidity": 60 }
    - Easy to debug, but larger size (inefficient for constrained IoT).
- **Binary formats (CBOR, Protocol Buffers, MessagePack)**
    - Compact, less readable for humans.
    - Example: instead of sending "temperature": 26.4, binary sends it as compressed bytes.
    - Advantage: saves **bandwidth** and **power**.

3. Messaging Protocols

- **MQTT (Message Queuing Telemetry Transport)**
    - Lightweight, based on Publish/Subscribe model.
    - Example: A sensor publishes data → broker → subscribers receive.
    - Ideal for **low-power IoT devices**.
- **CoAP (Constrained Application Protocol)**
    - REST-like protocol optimized for constrained devices.

    - Works over UDP, very lightweight.
    - Example: GET /sensor/temp from device.
- **HTTP/HTTPS**
    - Standard web communication, heavier than MQTT/CoAP.
    - Used in many cloud APIs, but consumes more power.

4. Getting on the Network

- IoT devices connect through different mediums:
    - **Wi-Fi** → high bandwidth, short range, power-hungry.
    - **Ethernet** → reliable, wired connection, not portable.
    - **Bluetooth LE** → short range, low power.
    - **LoRa/ZigBee** → long-range, low-power, very low data rate.
- Devices need **MAC address** + **IP address** for unique identification.

5. Traffic Generation & Analysis

- IoT devices generate **telemetry traffic** (sensor readings, logs).
- **Traffic patterns**: periodic (temperature every 5s) or event-driven (motion detected).
- Tools like **Iperf** test throughput and latency.
- **Wireshark** captures and analyzes packet details (to debug protocols like MQTT/CoAP).

6. IoT Cloud Models

- **IaaS (Infrastructure as a Service)** → Rent raw servers (AWS EC2, Azure VMs).
- **PaaS (Platform as a Service)** → IoT services & middleware (Azure IoT, AWS IoT).
- **SaaS (Software as a Service)** → Ready-to-use apps (Blynk, ThingsBoard dashboards).
- Trade-off: **Control vs Speed**. Startups often use SaaS/PaaS for faster development.

7. Device Identity

- Each device must have a **unique identity**:
  - **MAC Address** (hardware unique).
  - **Serial numbers** (assigned by manufacturer).
  - **Digital Certificates/Keys** (used for authentication with cloud).
- Purpose: **security** (ensures only trusted devices connect).

8. IoT Security Basics

- IoT devices are often targeted (e.g., **Mirai Botnet**).
- Key security principles:
  - **Encryption** (TLS/SSL → data is unreadable if intercepted).
  - **Authentication** (verify the device/user identity).
  - **Firmware updates** (patch vulnerabilities).
  - **Least-privilege principle** (device only gets minimal access).

9. Impact on Software & Hardware

- **Software**: must include optimized network stack (TCP/IP, MQTT libraries), error handling, retry logic.
- **Hardware**: needs enough **RAM/Flash** to handle networking libraries + secure crypto accelerators (for TLS).

Practical Activities

- Write an **MQTT publisher** on ESP32 (send temperature data to a broker).
- Use **Wireshark** to capture MQTT packets.
- Compare sending data in **JSON vs Binary** format.
- Create a simple **device identity system** with unique IDs.

Summary

- IoT networking connects devices → cloud → users.
- Data formats (JSON vs binary) impact efficiency.
- Protocols like **MQTT** and **CoAP** are preferred for IoT.
- Devices need identity + security to stay safe.
- Both **software and hardware** must be designed for secure, efficient networking.

References

- MQTT.org
- CoAP Technology
- https://youtu.be/OD2pxBN-MyI?si=hW-J4dNHhYSy_Zh_
- Microchip University Course