OSI Model and IoT Computer Communications

The OSI Model

The **Open Systems Interconnection (OSI)** model does not exist. It is a conceptual framework that standardizes the functions of a telecommunication system, regardless of the underlying technology. It helps designers and developers understand where the different technologies fit and how they interact.

The OSI model is the basis of real world communication stacks, such as the TCP/IP stack that powers the Internet and the BLE stack that is the foundation of Bluetooth Low Energy communications.

Each layer in the model serves a specific purpose and communicates only with the layers directly above and below it.

OSI Layers Overview

Layer	Name	Purpose	Example Protocols/Technologies
7	Application	Interface for end-user applications	HTTP, FTP, DNS, MQTT, CoAP
6	Presentation	Data formatting, encryption, compression	TLS, JPEG, ASCII, SSL
5	Session	Session management (open, maintain, close sessions)	NetBIOS, RPC
4	Transport	End-to-end connections, reliability, flow control	TCP, UDP
3	Network	Logical addressing and routing	IP, ICMP, OSPF
2	Data Link	Physical addressing, error detection	Ethernet, ARP, PPP, WiFi MAC layer
1	Physical	Transmits raw bitstreams over physical medium	Cables, RF, Bluetooth radio, 802.11

Spend a little time considering these layers. Do they make sense? What protocols or technologies do you already know about?

Myself, I always stumble a little when I see ASCII there. Why do you think that is? What is the logic of it being there after all?

Key Concepts per Layer

- **Layers 1–2 (Hardware-oriented)**: Handle bits and frames on a physical medium (e.g. WiFi, cables, BLE).
- **Layers 3–4 (Network stack)**: Govern addressing, routing, reliability, and packet sequencing.
- **Layers 5–7 (Application-facing)**: Handle data representation,

encryption, sessions, and application protocols.

Why It Matters

Even though most real-world systems (like TCP/IP) collapse or merge layers, understanding the OSI model gives clarity on where a problem might occur — e.g., is it routing (Layer 3), session persistence (Layer 5), or payload formatting (Layer 6)?

> Jorå, s'att vi hade ju problem me lejer 3-svirren men så visare sej att de bara var layer 7-brandisen som blockade anroppen från databasen.

(Probably a quote from a network tech guy. It makes a bit more sense when you know about the layers.)

In **IoT**, the layered model helps when:

- Choosing between **UDP vs TCP** for power efficiency
- Deciding whether **MQTT over TLS** is too heavy for an embedded device
- Debugging whether a sensor isn't transmitting due to RF interference (Layer 1) or IP misconfiguration (Layer 3)

Being able to communicate clearly around the OSI stack will give you a strong foundation for any IoT project and will impress on your peers. Fortune awaits.

IoT and Computer Communication Fundamentals

Internet of Things (IoT) devices often work in constrained environments. They must communicate efficiently, securely, and reliably, even with limited power, bandwidth, or processing capability.

Core Characteristics of IoT Communication

Feature	Description
Low power	Protocols and chips are optimized for battery life (e.g., 802.15.4, LoRa)
Low bandwidth	Communication may be kilobits per

	second or less
Intermittent links	Devices may sleep, move out of range, or rely on lossy wireless networks
Security-critical	Many IoT deployments are in sensitive environments (e.g. healthcare, home)

Typical Communication Architecture

A typical IoT setup includes:

1. **Sensor nodes / embedded devices**
   Communicate using lightweight protocols (MQTT, CoAP, BLE) via TCP/UDP over IP.
2. **Gateways**
   Aggregate data from sensors and act as **protocol translators** (e.g., BLE to WiFi, or ZigBee to Ethernet).
3. **Cloud or local servers**
   Store, analyze, and visualize sensor data; may issue commands to actuators.
4. **Control applications**
   Run on web dashboards, phones, or edge systems.

Common Protocol Stacks in IoT

Stack	Notes
TCP/IP	Most commonly used stack; interoperable with internet infrastructure
6LoWPAN	IPv6 over Low-Power Wireless Personal Area Networks; compresses headers
MQTT	Lightweight publish/subscribe protocol on top of TCP
CoAP	Constrained Application Protocol; REST-like, runs over UDP
Bluetooth LE / BLE	Often used for local, battery-efficient communication between mobile and IoT nodes

Transport Choices: TCP vs UDP in IoT

- **TCP**: Guarantees delivery and order, but has more overhead — used when reliability is essential (e.g., OTA firmware updates).
- **UDP**: Minimal overhead, fast, but unreliable — good for fast telemetry or sensor readings.

Example: Communication Flow in an IoT Thermostat

1. Thermostat collects temperature (Application layer).
2. Data is serialized and encrypted (Presentation layer).
3. Device establishes a session to the cloud (Session layer).
4. Transmits the data using TCP (Transport layer).
5. Uses IP addressing and routing (Network layer).
6. Frames the data for WiFi (Data Link layer).
7. Transmits bits via RF (Physical layer).

> I had planned on doing a sketch for this, it would be very pretty and colorful. Can you do it for me? Send it to me on Teams and I'll include it here.

Suggested Practice

1. **OSI Mapping Exercise**: Given a packet dump in Wireshark, identify which layers are visible.
2. **Protocol Stack Tracing**: Choose an IoT device (e.g., ESP32) and map out its communication stack from physical to application layer.
3. **Protocol Comparison Table**: Compare MQTT, CoAP, and HTTP in terms of OSI layers used, overhead, and typical use cases.
4. **Debugging Drill**: Diagnose a simulated IoT device that "won't connect" — identify if the failure is at Layer 1, 3, or 7.

Quiz Questions

1. What are the 7 layers of the OSI model and the primary function of each?
2. Which OSI layer is responsible for routing packets between networks?
3. What advantages does UDP offer over TCP for constrained IoT devices?
4. How does CoAP differ from MQTT in terms of transport layer usage?
5. What are some real-world technologies that operate at Layer 1 and

Layer 2 in IoT systems?

6. Why is it important for IoT devices to minimize communication overhead?
7. In the OSI model, which layer would be responsible for handling TLS encryption?
8. What are the trade-offs between using WiFi and BLE in an IoT deployment?
9. Explain how a gateway can act as a protocol translator in an IoT system.
10. What might cause a Layer 3 (Network) failure in a wireless IoT deployment?