IOT/M2M SYSTEMS LAYERS AND DESIGN STANDARDIZATION

1. Physical cum Data-Link Layer (Layer 1 & 2) – Gathering Data

This is the foundation of IoT communication, where data is physically transmitted between devices.

Key Functions:

- **Physical layer:** Handles hardware connections like wired (Ethernet, Fiber Optic) or wireless (Wi-Fi, Bluetooth, Zigbee, LoRa, 5G).
- **Data-Link layer:** Ensures reliable point-to-point communication using MAC (Media Access Control) and LLC (Logical Link Control).

Example: Sensors detecting temperature and sending raw data over Zigbee.

2. Data Adaptation Layer (Layer 2) – Enriching Data

This layer refines raw data before it moves up the network.

Key Functions:

- Converts different data formats for compatibility.
- Adds error detection and correction mechanisms.
- Handles device addressing via MAC addresses.

Example: A smart meter adapts electrical readings to a suitable format for transmission.

3. Network Layer (Layer 3) – Routing Data

This layer is responsible for determining the best path for data transmission across networks.

Key Functions:

- Assigns IP addresses to IoT devices.

- Uses routing protocols (IPv6, MQTT, CoAP) for efficient data delivery.
- Ensures data reaches the correct cloud server or gateway.

Example: A smart irrigation system sends soil moisture data to a cloud server via a Wi-Fi router.

4. Transport Layer (Layer 4) – Streaming Data

This layer ensures secure and reliable data transmission between IoT devices and cloud platforms.

Key Functions:

- Provides error control & retransmission if data is lost.
- Uses TCP (Transmission Control Protocol) for reliability.
- Uses UDP (User Datagram Protocol) for real-time, low-latency communication.

Example: A live video stream from a security camera uses UDP for low-latency streaming.

5. Session Layer (Layer 5) – Managing Connections

This layer establishes, maintains, and terminates communication sessions between IoT devices.

Key Functions:

- Synchronizes multiple data streams.
- Manages authentication and re-connection in case of failures.

Example: A smart home assistant (like Alexa) maintains a session with a cloud service while processing voice commands.

6. Presentation Layer (Layer 6) – Formatting and Securing Data

This layer translates data into a readable format and ensures security.

Key Functions:

- Converts data between different formats (JSON, XML, binary).
- Implements encryption (TLS/SSL) for secure transmission.
- Compresses data for efficient transfer.

Example: A smart fridge encrypts temperature logs before sending them to the cloud.

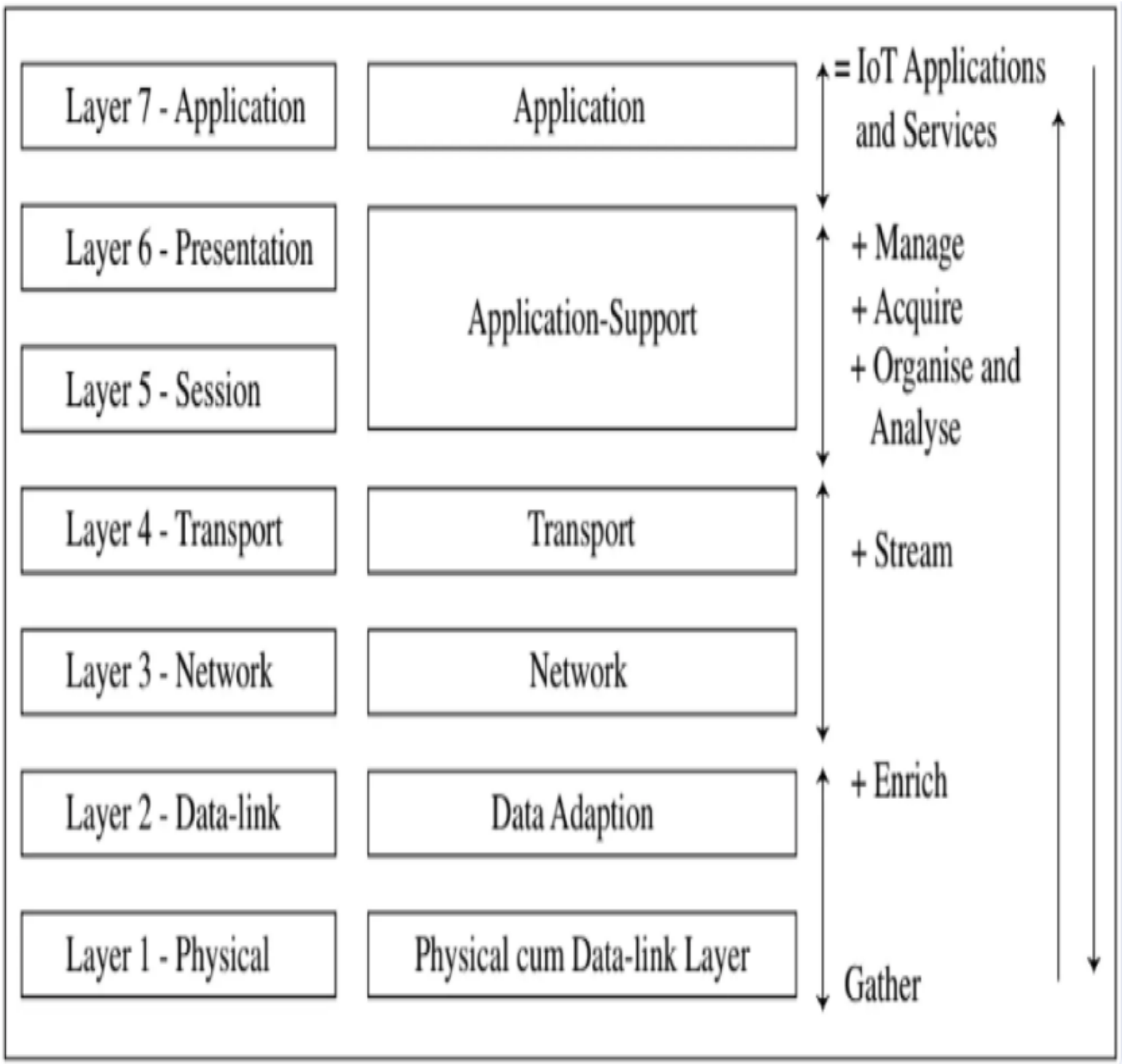
7. Application Layer (Layer 7) – Managing IoT Applications & Services

This is the topmost layer where users interact with IoT applications through web dashboards or mobile apps.

Key Functions:

- Data acquisition, storage, and analytics.
- Provides interfaces (REST APIs, MQTT brokers) for developers.
- Displays insights on dashboards or mobile apps.

Example: A smart city traffic management system collects real-time traffic data and shows it on a dashboard for city planners.



Summary:

Layer	Function	Example
Physical cum Data-Link	Sensing and transmitting raw data	Temperature sensors
Data Adaptation	Formatting and refining data	Error correction in sensor data
Network	Routing and IP addressing	Sending data via Wi-Fi or LPWAN
Transport	Ensuring reliable transmission	Secure data exchange with TCP/UDP
Session	Managing connections	Smart assistant interactions
Presentation	Data security and conversion	Encryption for IoT data
Application	Providing insights and control	IoT dashboards & mobile apps

Design Standardization

Standardization ensures consistency, security, and compatibility in IoT/M2M systems. Key standardization bodies, frameworks, and protocols include:

International Standards

- ISO/IEC 30141 **– IoT Reference Architecture.**
- IEEE P2413 **– IoT Architectural Framework.**
- ITU-T Y.2060 **– Overview of IoT and its framework.**

Communication Protocols

- **MQTT (Message Queuing Telemetry Transport) –** Lightweight messaging protocol for IoT.

- **CoAP (Constrained Application Protocol) –** Web-based communication for constrained devices.
- **HTTP/HTTPS –** Standard protocols for web-based IoT applications.
- **DDS (Data Distribution Service) –** High-performance real-time communication for industrial IoT.
- **AMQP (Advanced Message Queuing Protocol) –** Middleware messaging for IoT applications.

Wireless Connectivity Standards

- **Zigbee and Z-Wave –** Wireless communication standards for smart home devices.
- **LoRaWAN and NB-IoT –** Low-power, wide-area network (LPWAN) protocols for IoT connectivity.
- **5G and LTE-M –** Cellular network standards for high-speed IoT applications.

- **Wi-Fi HaLow (802.11ah) –** Low-power Wi-Fi standard for IoT devices.

Industrial and Security Standards

- **OPC UA (Open Platform Communications Unified Architecture) –** Standard for industrial IoT (IIoT) communication.
- **NIST Cybersecurity Framework –** Guidelines for securing IoT devices.
- **IEC 62443 –** Industrial cybersecurity framework for IoT and M2M.
- **FIDO (Fast Identity Online) –** Authentication standard for IoT security.

Middleware and Edge Computing Frameworks

- **OneM2M –** Common service layer for M2M communication.
- **EdgeX Foundry –** Open-source framework for edge computing in IoT.
- **OpenFog Consortium –** Fog computing architecture for IoT.

By adhering to these standards, IoT systems achieve better security, interoperability, scalability, and reliability, ensuring seamless integration across diverse devices and platforms.