Supplemental Reading for IoT Data Transfer Protocols

IoT Data Transfer Protocols

In this reading, you will learn how Internet of Things (IoT) devices send and receive data across networks. As an IT Support specialist, you may need to support data collection from IoT devices. For example, you may work for a company that uses an array of IoT sensors in a manufacturing setting to help with the remote monitoring and proactive maintenance of industrial machines. You may need to manage the software applications and data transfer protocols that support automated and human interaction with the IoT devices and the data they collect.

Data protocol models used with IoT

There are two common data protocol models to illustrate how low-power IoT devices share data:

- **Request/Response model:** Often used in distributed systems where the communication flow between servers and clients consists of requests and responses for data. Examples include HTTP and CoAP (described in the “IoT data protocols at the application layer” section below)
- **Publish/Subscribe model:** A framework for message exchanges between publishers (hosts) and subscribers (clients) that are routed through a broker. Subscribers can sign up to a channel to receive notices through the broker when the publisher releases new messages. Examples: MQTT and AMQP (described in the “IoT data protocols at the application layer” section below).

IoT data protocols at the application layer

IoT devices can collect environmental data around their physical location (e.g., temperature), equipment data (e.g., maintenance status), and metered data (e.g., electricity usage). Data protocols are needed to transfer and format the data for use by applications that interface with either humans or automated systems. IoT devices can be configured to use various data transfer and formatting protocols at the OSI application/software layer of communication.

Most IoT devices can use at least one of the following data transfer protocols:

- **HyperText Transfer Protocol / Secure (HTTP/HTTPS):** HTTP and HTTPS are the most widely used information transfer protocols across the World Wide Web (WWW). The protocols define how information is formatted and transmitted. HTTP/HTTPS uses ASCII formatting, has a header size of 8 bytes, and is designed for transmitting documents. HTTP/HTTPS use either Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) for sending information across the internet. HTTP/HTTPS uses the request/response model. When a website address is entered into a browser, HTTP/HTTPS sends a request to the site’s web server, which then returns an HTTP/HTTPS formatted response to the browser. The protocols use ports 80 or 8080 and data security is provided on the HTTPS version of the protocol. HTTP is supported by Google Cloud IoT Core for device-to-cloud communication.
- **Machine-to-Machine (M2M) Communication Protocols:** A set of direct communication methods for low-power devices, machines, and systems. There are three primary architectural and protocol groups in M2M electronic communications:
  - **Representational State Transfer (REST):** An architectural style for communication amongst web accessible systems.
  - **Service-oriented Architectures (SOA):** An architecture for data exchanges in industrial automation systems.
  - **Message Oriented Protocols:** A protocol for asynchronous data transfers for distributed systems.
- **Message Queue Telemetry Transport (MQTT):** An IoT data-centric interaction protocol for M2M that uses a simple publish-subscribe model. MQTT supports Quality of Service (QoS), uses TCP for sending information, and utilizes Secure Sockets Layer (SSL) and Transport Layer Security (TLS) for security. MQTT using binary format and 2-byte header sizes for efficient messaging. MQTT is supported by Google Cloud IoT Core for device to cloud communication.
- **Constrained Application Protocol (CoAP):** A web transfer protocol for IoT constrained nodes and networks designed for M2M applications. CoAP is used for IoT applications like building automation and smart energy management. CoAP is very similar to HTTP: both are based on the REST model and both place resources on a server that is accessible to clients via a URL.
- **Advanced Message Queuing Protocol (AMQP):** An open standard for messaging amongst applications in different organizations and/or platforms. Its purpose is to remove vendor lock-in for app communication. In addition to interoperability, AMQP also offers reliability and security.
- **Extensible Messaging and Presence Protocol (XMPP):** A decentralized, open standard for chat, messaging, video and voice calls, collaboration tools, and more. Built upon Japper, XMPP offers a proven communication technology that is extensible, flexible, and diverse.
- **Data Distribution Service (DDS):** An API standard and middleware protocol from the Object Management Group. Middleware exists in the OSI applications layer, between software and the operating system. DDS uses the publish-subscribe communications model. DDS is also data-centric, provides low-latency data connectivity, and helps the devices in an IoT ecosystem share data more efficiently. DDS is reliable, scalable, and provides control of QoS parameters, including bandwidth and resource limits.

Mark as completed