Chapter 7a: Cloud Protocols: Summary

Time ¼ Hour

This chapter will help you understand “the Cloud” and the basics of cloud protocols, which are also called application layer protocols. These protocols include HTTP, MQTT, AMQP, and CoAP. This chapter provides a summary of the protocols. Chapters 7B and 7C will cover HTTP and MQTT, respectively, in greater detail. AMQP and CoAP will be covered in a future revision of this material.

Table of Contents

Document conventions

Convention	Usage	Example
`Courier New`	Displays code	`CY_ISR_PROTO(MyISR);`
Italics	Displays file names and paths	*sourcefile.hex*
[**bracketed, bold**]	Displays keyboard commands in procedures	[**Enter**] or [**Ctrl**] [**C**]
Menu > Selection	Represents menu paths	**File > New Project > Clone**
Bold	Displays commands, menu paths and selections, and icon names in procedures	Click the **Debugger** icon, and then click **Next**.

7a.1 "The Cloud"

What is the Cloud? The Cloud is a simple name for a giant amalgamation of all the stuff that you need in order to provide websites and other network-based services (e.g. iTunes). Why do you need the Cloud? When you try to service large numbers of people and devices you have a very difficult and expensive problem. To have a fast and always available system, you need to have enough networks, disk drives, computers, and people to run it all. The solution to this problem is a standardized, shared, scalable system: The Cloud.

The term "The Cloud" generally includes:

- Networks (high bandwidth, worldwide, distributed)
- Storage (disks, databases)
- Servers (running Windows and Unix)
- Security
- Scalability
- Load Balancing
- Fault tolerance (redundancy)
- Management tools (reports, user identity management, etc.)
- Software (Web servers, APIs, Languages, Development tools etc.)

To make something interesting you need to be able to hook up your IoT device(s) (the "T" in IoT) to the Cloud (the "I" in IoT) which is the goal of this chapter.

7a.2 Application Layer Protocols

How do you get data to and from the Cloud? Simple, there are a number of standardized application layer protocols to do that task.

7a.2.1 Hyper Text Transfer Protocol (HTTP)

HTTP (https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) is a text-based Application Layer Protocol that operates over TCP Sockets. It can perform the following functions:

- GET (retrieve data) from a specific place
- POST (put data) to a specific place
- HEAD, PUT, DELETE, TRACE, OPTIONS, CONNECT, PATCH (less commonly used)

To initiate these commands, you open a socket typically to TCP port 80, send the text-based command (CRLF terminated) and read the replies. This request/reply protocol is used for every command. Replies are sent with a resulting Content-Type string which indicates the type of data encoding for the response. The content-type string uses a Multipurpose Internet Mail Extension (MIME) type to indicate the type of data being received (e.g. text/html or image/jpeg).

For instance, you can send an HTTP GET request followed by the Host header to open "/index.html" on www.example.com:

```
GET /index.html HTTP/1.1
Host: www.example.com
```

www.example.com will respond with:

```
HTTP/1.1 200 OK
Date: Mon, 23 May 2005 22:38:34 GMT
Content-Type: text/html; charset=UTF-8
Content-Encoding: UTF-8
Content-Length: 138
Last-Modified: Wed, 08 Jan 2003 23:11:55 GMT
Server: Apache/1.3.3.7 (Unix) (Red-Hat/Linux)
ETag: "3f80f-1b6-3e1cb03b"
Accept-Ranges: bytes
Connection: close
<html>
<head>
<title>An Example Page</title>
</head>
<body>
Hello World, this is a very simple HTML document.
</body>
</html>
```

Chapter 7a: It is possible (and semi-common) to build IoT devices that use HTTP to "PUT" their data to webservers in the Cloud and "GET" their instructions/data from webservers. However, HTTP is somewhat heavy (that is, bandwidth intensive) and is generally being displaced by other protocols that are more suited to IoT.

7a.2.2 Message Queueing Telemetry Transport (MQTT)

MQTT (https://en.wikipedia.org/wiki/MQTT) is a lightweight messaging protocol that allows a device to **Publish Messages** to a specific **Topic** on a **Message Broker**. The Message Broker will then relay the message to all devices that are **Subscribed** to that **Topic**.

The format of the messages being sent in MQTT is unspecified. The message broker does not know (or care) anything about the format of the data and it is up to the system designer to specify an overall format of the data. All that being said, JavaScript Object Notation (JSON) has become the lingua franca of IoT.

MQTT operates on TCP Ports 1883 for non-secure and 8883 for secure (TLS).

Cloud providers that support MQTT include Amazon AWS and IBM Bluemix.

Topic

A Topic is simply the name of a message queue e.g. "mydevice/status" or "mydevice/pressure". The name of a topic can be almost anything you want but by convention is hierarchical and separated with slashes "/".

7a.2.2.1 Publish

Publishing is the process by which a client sends a message as a blob of data to a specific topic on the message broker.

7a.2.2.2 Subscription

A Subscription is the request by a client device to have all messages published to a specific topic sent to it.

7a.2.2.3 Message Broker

A Message Broker is just a server that handles the tasks:

- Establishing connections (MQTT Connect)
- Tearing down connections (MQTT Disconnect)
- Accepting subscriptions to a Topic from clients (MQTT Subscribe)
- Turning off subscriptions (MQTT Unsubscribe)
- Accepting messages from clients and pushing them to the subscribers (MQTT Publish)

7a.2.2.4 QOS

MQTT provides three levels of Quality of Service (QOS):

- Level 0: At most once (each message is delivered once or never)
- Level 1: At least once (each message is certain to be delivered, possibly multiple times)
- Level 2: Exactly once (each message is certain arrive and do so only once)

7a.2.3 Advanced Message Queuing Protocol (AMQP)

AMQP (https://en.wikipedia.org/wiki/Advanced_Message_Queuing_Protocol) is a binary application layer protocol designed to efficiently support a wide variety of messaging applications and communication patterns. It provides flow controlled, message-oriented communication with message-delivery guarantees such as *at-most-once*, *at-least-once* and *exactly-once*, and authentication and/or encryption based on SASL and/or TLS. It assumes an underlying reliable transport layer protocol such as Transmission Control Protocol (TCP).

The AMQP specification is defined in several layers: (i) a type system, (ii) a symmetric, asynchronous protocol for the transfer of messages from one process to another, (iii) a standard, extensible message format and (iv) a set of standardized but extensible 'messaging capabilities.'

Cloud providers that use AMQP include Microsoft (e.g. Windows Azure), VMWare, and Redhat.

7a.2.4 Constrained Application Protocol (CoAP)

CoAP (https://en.wikipedia.org/wiki/Constrained_Application_Protocol) makes use of two message types, requests and responses, using a simple, binary, base header format. The base header may be followed by options in an optimized Type-Length-Value format. CoAP is by default bound to UDP and optionally to DTLS, providing a high level of communications security.

Any bytes after the headers in the packet are considered the message body, if any. The length of the message body is implied by the datagram length. When bound to UDP the entire message MUST fit within a single datagram. When used with 6LoWPAN as defined in RFC 4944, messages SHOULD fit into a single IEEE 802.15.4 frame to minimize fragmentation.

The mapping of CoAP with HTTP is also defined, allowing proxies to be built providing access to CoAP resources via HTTP in a uniform way.

Cloud providers that use CoAP include Samsung ARTIK.

7a.3 Further Reading

- [2] RFC2045 – "Multipurpose Internet Mail Extensions"; Internet Engineering Task Force (IETF) - https://tools.ietf.org/html/rfc2045
- [4] RFC2616 – "Hypertext Transfer Protocol (HTTP) " ; Internet Engineering Task Force (IETF) - https://tools.ietf.org/html/rfc2616
- [5] RFC7159 – "The Javascript Object Notation (JSON) Data Interchange Format"; Internet Engineering Task Force (IETF) - https://tools.ietf.org/html/rfc7159
- [6] MQTT - http://mqtt.org/
- [7] RFC7959 – "The Constrained Application Protocol (CoAP)" ; Internet Engineering Task Force (IETF) - https://tools.ietf.org/html/rfc7252
- [8] AMQP - http://www.amqp.org/