# 第7a章：云连接的应用层协议

## 目标

学习第七章后，你将了解如何用某种云应用协议（MQTT、COAP、AMQP、HTTP或Sockets）来构建完整的WICED IoT　App。此外，你还将全面了解以上各种协议。我们还将介绍Amazon AWS、IBM Bluemix、阿里云和Microsoft Azure等云厂商。

本部分包含了解具体云协议有关章节所需的基础知识。通过本节学习，你将基本了解HTTP、MQTT、AMQP和COAP等应用层通讯协议。

## 时间：3个半小时

## 基础知识

### “云”

什么是云？云是提供网站和其他网络服务（如iTunes）所需的所有事项的庞大组合的统一称谓。为什么需要云？要服务大量人群和设备，就会面临非常艰巨且成本高昂的问题。为了获得快速且随时可用的系统，必须拥有足够的网络、磁盘驱动器、计算机和人员（这样才能保持运行）。这个问题的解决方案就是采用标准化的、共享的可扩展系统：云。

“云”这个术语通常包括：

1. 网络（高带宽、全球化、分布式）
2. 存储（磁盘、数据库）
3. 服务器（运行Windows和Unix）
4. 安全性
5. 可扩展性
6. 负载均衡
7. 故障容差（冗余）
8. 管理工具（报告、用户身份管理等）
9. 软件（Web服务器、API、语言、开发工具等）

为了提供有意义的服务，你应当让IoT设备（也就是IoT中的“T”）连接到云（即IoT中的 “I”），这就是本章的目的所在。

### 应用层协议

如何向云端传输数据？如何从云端获得数据？很简单，一系列标准化应用层协议能帮助你实现这一目的。

[超文本传输协议（HTTP）](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol)

HTTP是一个通过TCP Sockets运行的基于文本的应用层协议。它执行以下功能：

* 从特定位置GET（检索数据）
* 向特定位置POST（放置数据）
* HEAD, PUT, DELETE, TRACE, OPTIONS, CONNECT, PATH（较少使用）

要启动以上这些命令，打开socket，通常是TCP端口80，发送文本命令（CRLF结尾）并读取答复。这种请求/应答协议可用于所有命令。答复运用结果内容类型（Content-Type）字符串发送，表明答复的数据编码类型。内容类型字符串采用Multipurpose Internet Mail Extension（MIME）类型来表明正在接收数据的类型（如text/html或 image/jpeg）。

举例来说，你可发送HTTP GET请求来打开[www.example.com](http://www.example.com)网站上的“/index.html”：

* GET /index.html HTTP/1.1

Example.com将做出以下回复：

* 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>

用HTTP来构建IoT设备也是可能的，不过不太常用。将数据**“放在”**云端的Web服务器，并从Web服务器**“GET”**指令/数据。不过，HTTP比较笨重（也就是说带宽占用较高），通常我们要用其他更适合IoT的通讯协议来替代。

[Message Queueing Telemetry Transport (MQTT，消息队列遥测传输)](https://en.wikipedia.org/wiki/MQTT)

MQTT是一种轻量级消息通讯协议，能让设备向消息代理（Message Broker）上的特定主题（**Topic）**发布消息（Publish Messages）。消息代理会向订阅该主题的所有设备分程传递该消息。

MQTT发送的消息格式是未经指定的。消息代理不知道（也不关心）数据格式，系统设计人员应指定数据整体格式。正因如此，[JavaScript Object Notation (JSON)](https://en.wikipedia.org/wiki/JSON) 已成为IoT的通用数据交换格式。

主题就是消息队列的名称，如“mydevice/status”或“mydevice/pressure”。主题名称几乎可以是任意的，不过通常采用“/”分级和隔开。

发布（publish）是指客户端将消息作为数据对象发送给消息代理的特定主题。

订阅(subscribe)是指设备请求获得发布到客户端特定主题的所有消息。

消息代理就是处理以下任务的服务器：

* 建立连接（MQTT连接）
* 断开连接（MQTT断连）
* 接受客户端主题订阅（MQTT订阅）
* 取消订阅（MQTT退订）
* 从客户端接受消息并将消息推送给订阅者（MQTT发布）

MQTT提供三级服务质量（QOS）：

* 0级：最多一次（每条消息只发送一次或不发送）
* 1级：至少一次（每条消息肯定发送，或许会多次发送）
* 2级：刚好一次（每条消息肯定会抵达，而且只会抵达一次）

非安全情况下MQTT工作在TCP端口1883上，安全情况下（TLS）工作在8883上。

支持MQTT的云提供商包括Amazon AWS和IBM Bluemix。

[Constrained Application Protocol (CoAP， [约束应用协议](https://en.wikipedia.org/wiki/Constrained_Application_Protocol))](https://en.wikipedia.org/wiki/Constrained_Application_Protocol)

[约束应用协议](https://en.wikipedia.org/wiki/Constrained_Application_Protocol)CoAP使用两种消息类型：请求和响应，并采用简单的二进制基础标题格式。基础标题后面可以是优化的类型-长度-值（Type-Length-Value）格式的选项。CoAP默认绑定UDP，可选[DTLS](https://en.wikipedia.org/wiki/DTLS)，提供了高级通信安全性。

数据包标题后面的所有字节都被视为消息正文。消息正文的长度受限于数据报长度。绑定于YDP时，整条消息**必须**包含在单一数据报中。如根据[RFC 4944](https://tools.ietf.org/html/rfc4944)定义与[6LoWPAN](https://en.wikipedia.org/wiki/6LoWPAN)配合使用，消息**应**包含在单一[IEEE 802.15.4](https://en.wikipedia.org/wiki/IEEE_802.15.4)帧中，以尽可能减少碎片化。

CoAP与[HTTP](https://en.wikipedia.org/wiki/HTTP)的映射也得到定义，能构建代理，以统一方式通过HTTP访问CoAP资源。

使用CoAP的云提供商包括Samsung ARTIK。

Advanced Message Queuing Protocol[（AMQP](https://en.wikipedia.org/wiki/Advanced_Message_Queuing_Protocol)，高级消息队列协议）

AMQP是一种二进制应用层协议，能高效支持各种不同的消息应用和通信模式。它提供流控制的消息导向通信，确保*最多一次*、*至少一次*和*刚好一次*的消息发送，而且提供基于[SASL](https://en.wikipedia.org/wiki/Simple_Authentication_and_Security_Layer)和/或[TLS](https://en.wikipedia.org/wiki/Transport_Layer_Security)的认证和/或加密功能。它假定传输控制协议（TCP）等底层传输层协议是可靠的。

AMQP规范定义涉及不同的层面：(i) 类型系统；(ii) 用于不同进程间用于传递消息的对称异步协议；(iii)标准的可扩展消息格式；(iv) 一系列标准化但可扩展的‘消息功能’。

使用AMQP的云提供商包括微软（如Windows Azure）、VMWare和Redhat。

### [JavaScript Object Notation（JSON](https://en.wikipedia.org/wiki/JSON)， javaScript对象符号）

JSON是一款开放式标准格式，使用可读取的文本来发送包含属性–值对的数据。JSON支持以下数据类型：

* 双精度浮点
* 字符串
* 布尔（真或假）
* 阵列（用“[]”指定阵列，值用“,”隔开）
* 键/值（键映射）对，也就是“键”:值（用“{}”指定键映射），用“,”将对隔开

键/值的值可以是阵列，也可以是键/值映射。

阵列可包含键/值映射。

举例来说，合法的JSON文件如下：

{

“astringkey” : “alan”,

“age” : 48,

“badass” : true,

“children”: [“Anna”,”Nicholas”],

“address” : {

“number”:”201”

“street”: “East Main Street”

“city”: “Lexington”,

“state”:”Kentucky”,

“zipcode”:40507

}

}

WICED SDK内置JSON解析器。你可在目录“Utilities🡪JSON\_parser”下找到相关函数。

## 额外阅读材料

[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/>