第 5 章

CHAPTER 5

连接与协议

连接是物联网的基本特性之一，也是物联网操作系统的基本需求之一。通信技术对于物联网的连接来说十分常用且关键，通信技术是互联网各单位之间进行信息传输和交流的物质基础。没有通信技术，物联网就不能“联”，也就不能构成“网”。无论是近距离无线传输技术还是移动通信技术，都影响着物联网的发展。

物联网涵盖了广泛的行业和用例，从单一受限设备扩展到嵌入式技术和云系统的大规模跨平台部署，实时连接。将所有这些结合在一起的是众多传统和新兴的通信协议，它们允许设备和服务器以新的、更加互连的方式相互通信。与此同时，数十个组织和联盟正在形成，希望统一破碎的有机物联网景观。图 5-1 显示了物联网连接示意。

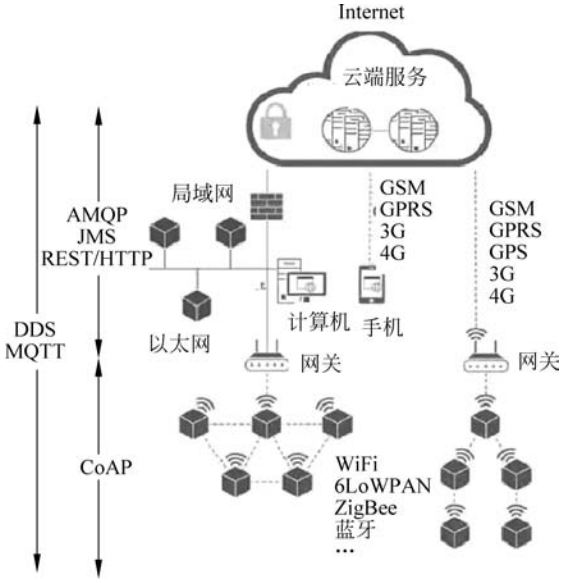

图 5-1　物联网连接示意图

物联网的通信环境有以太网、WiFi、RFID、NFC（近距离无线通信）、ZigBee、6LoWPAN（IPv6 低速无线版本）、蓝牙、GSM、GPRS、GPS、3G、4G 等网络，而每一种通信应用协议都有一定适用范围。

针对物联网的特点，低功耗广域网和低功耗局域网应运而生。目前，全球范围内主要形成两大技术阵营，在智能家居、工业的数据采集等局域网的场合下，一般采用的是短距离通

信技术,如 ZigBee 和蓝牙 4.0。对于范围较广,距离较远的连接,就会采用远距离的通信技术,包括 LoRa、NB-IoT、Sigfox、Weightless,这些技术虽然特点各异,但都能满足物联网低成本、低功耗、大量连接的需求。

低功耗广域网(Low-Power Wide-Area Network,LPWAN)是为了满足越来越多远距离物联网设备的连接要求,即为了物联网应用中的 M2M(Machine to Machine,机器对机器)通信场景而优化、发展的一项新技术。低功耗广域网具有低功耗、远距离、低运维成本等特点,可以真正实现大区域物联网低成本全覆盖,在未来的智慧城市的建设发展过程中,低功耗广域网的应用将会越来越多。根据是否授权,低功耗广域网又分为两个分支:使用授权频段与使用非授权频段的低功耗广域网,使用授权频段的是 NB-IoT、LTE-M 等,使用非授权频段的是 Sigfox、LoRa 等。

低功耗局域网也是多种技术并存,例如 ZigBee、蓝牙 4.0、WiFi 等技术,每个技术都有自己的优势和领域。相对于低功耗广域网,低功耗局域网具有产业链成熟、应用普及、稳定性强等优点,能够满足物联网应用的需要。

物联网操作系统对于低功耗物联网通信技术的需求与支持力度都很大。比如华为 LiteOS 支持的接入方式就包括 NB-IoT、2G、3G、4G 等。

在第 1 章介绍的物联网的 3 个层次中,感知层和应用层中的信息或数据需要通过传输层进行通信。目前 TCP/IP 已经成为互联网事实上的标准协议,而物联网通信协议主要也是运行在互联网 TCP/IP 之上的设备通信协议,负责设备通过互联网等各类网络实现通信和数据交换。

通信协议与通信技术及相关硬件的关联度较高,现有的产品标准尚未完全确立,因此可以预见到未来的物联网接入所采用的数据通信协议仍将是百花齐放的格局。目前应用层的网络通信协议主要有 HTTP、MQTT、CoAP、XMPP 等,通过这些协议实现系统与系统、物与物之间的信息交换。

以华为物联网平台为例,华为物联网平台目前支持设备采用协议包括 MQTT、CoAP、LwM2M 等,如表 5-1 所示。

表 5-1 华为物联网平台支持的主要协议

通信协议	协议描述	应用场景
LwM2M	LwM2M 是开发移动联盟 OMA 定义的用于设备管理的应用层通信协议,主要是用在资源受限的嵌入式设备上	NB-IoT 设备接入平台,适合业务实时性要求不高、功耗低、信号覆盖广的场景
CoAP	CoAP 是资源受限设备和受限网络专用的 Web 传输协议,专为机器对机器的应用而设计。CoAP 提供请求/响应交互模型,支持内置的服务和资源发现。需要底层实现 UDP 协议	NB-IoT 设备接入平台,适合业务实时性要求不高、功耗低、信号覆盖广的场景
MQTT	MQTT 是一种物联网连接协议,提供非常轻量级的发布/订阅消息传输方式,用于在低带宽、不可靠的网络的设备管理。该协议构建于 TCP/IP 协议上	对设备的可靠性和实时性要求高,适合长连接的场景,如智能路灯等

MQTT 是安全的基于 TLS 的加密协议。采用 MQTT 协议接入平台的设备，设备与物联网平台之间的通信过程，数据都是加密的，具有一定的安全性。MQTT 应用于计算能力有限，且工作在低带宽、不可靠的网络的远程传感器和控制设备，适合长连接的场景，如智能路灯等。

再比如 AliOS Things。AliOS Things 支持丰富的云端连接协议：

- Alink——阿里巴巴云平台，适用于智能生活；也包括 WiFi 配置组件 YWSS。
- MQTT——标准 MQTT 协议；已与阿里巴巴云物联网套件良好结合。
- CoAP——基于 UDP 的轻量级协议。和 CoAP FOTA 结合便可为 NB-IoT 设备建立一个只有 UDP 的系统。

本章首先介绍 NB-IoT 接入技术和 LoRa 技术，然后介绍 LiteOS 支持的 MQTT 协议、CoAP 协议和 LwM2M 协议。

5.1 NB-IoT

近年来，随着通信技术的不断发展，物联网技术应用也进一步地提上各国发展日程，工业 4.0、智能城市、智能家居、智慧农业等方向蓬勃发展。NB-IoT(Narrow Band Internet of Things，基于蜂窝的窄带物联网)通信技术，作为现阶段全世界发展最快的无线物联网通信技术，其广域传输的技术特点是真正实现万物互联的关键技术，近年来受到电信运营商、物联网设备制造商、物联网基站制造商以及传感器制造商等业内人士的广泛重视。

NB-IoT 通信技术有广域连接、低成本、低能耗、高稳定性等优势，再加之我国三大电信运营商对网络的不断优化，使其更加安全，所以近年来大多智慧农业、智慧城市等应用相关领域都在加速 NB-IoT 的设备部署以及应用。图 5-2 显示了 NB-IoT 系统与网络体系架构。

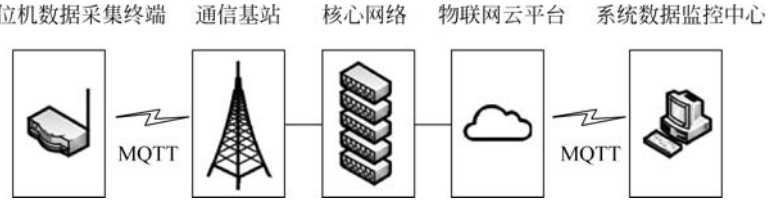

图 5-2 NB-IoT 系统与网络体系架构

如图 5-2 所示，NB-IoT 系统与网络体系架构基本分为以下 5 个部分：下位机数据采集终端、通信基站、核心网络、物联网云平台和系统数据监控中心。本节主要介绍 NB-IoT 技术。

5.1.1 NB-IoT 的技术特点

NB-IoT 的诞生大大推进了物联网发展的脚步一跃成为全球热点，前景广阔。NB-IoT 的技术特点体现在以下几个方面。

1. NB-IoT 网络覆盖强

NB-IoT 设计标准中物联网通信 MCL(Minimum Coupling Loss，最小耦合路径损耗)定为 164dB，在 GSM 网络覆盖基础上强度提升 20dB。信号覆盖分为上行与下行信号增强

技术。

上行信号增强技术：虽然 NB-IoT 设备终端上行发射功率(23dBm)比传统 GSM(33dBm)终端上行发射功率低，但是 NB-IoT 上行信号通过缩窄传输带宽从而提高上行信号发射功率谱的密度提升信号覆盖，同时增加上行数据重复发送次数，数据最大重复次数可达 2048 次，从而提升信号覆盖。

下行信号增强技术：下行 NB-IoT 信号比 GSM 信号发射功率大，从而提升覆盖强度，另外三大运营商搭建基站以高发射率向终端传输信息，比终端高很多的发射率可以进一步提升信号的覆盖。从网络覆盖范围来看，NB-IoT 的信号覆盖半径是 GSM/LTE 的 4 倍。NB-IoT 信号覆盖能力的提升提高了物联网终端设备的通信效率。

2. NB-IoT 低功耗运行

在 NB-IoT 标准制定之初，3GPP(the 3rd Generation Partnership Project，第三代合作伙伴) 制定的目标电池寿命为 10 年，所以 NB-IoT 标准中引入了 PSM(Power Saving Mode，省电模式)和 eDRX(extended Discontinuous Reception，增强型非连续性接收)模式。

PSM 模式是 3GPP R12 中引入的一种节能状态，凡是支持 PSM 模式的物联网终端在进入空闲模式一段时间后，就会切换至 PSM 模式；此时物联网终端模块的 PA(射频)部分停止工作，终端模块 AS(接入层)部分相关功能停止工作从而减少射频、信号处理等部分功耗，以达到低功耗的目的。由于设备进入 PSM 模式后射频停止工作，终端模块便无法接收到服务器发来的信息，数据信息无法发送至终端模块。当从 PSM 模式唤醒后，也就可以直接收发数据。

eDRX 模式是 3GPP R13 中引入的一种节能状态，其是在 DRX(Discontinuous Reception，不连续接收模式)模式技术上加以扩展，得以支持更长周期的寻呼。将 DRX 技术用于 IDLE 中进行周期性监听寻呼，如有通信请求可以立即唤醒，如果没有可以降低终端的功耗。

NB-IoT 通过使用 PSM 和 eDRX 技术以达到高效的数据传输与高效的能耗控制，从而使 NB-IoT 终端设备拥有更好的场景适应能力。

3. NB-IoT 低成本

NB-IoT 模块设计之初就本着低功耗、低带宽、低速率的需求按照 NB-IoT 标准进行设计，从而达到低成本优势。在上述需求中，低传输速率使得传输模块不需要较大的缓存空间，模块低功耗代表着模块射频(Radio Frequency，RF)设计要求较低，数据传输低带宽说明模块不需要较为复杂的均衡算法，从而大大简化了盲检次数、调制解调的编码，缩小了传输块等，简化了天线模块设计。这样就使得 NB-IoT 芯片模块相对 LTE 芯片模块设计大幅简化，进而较其他物联网芯片模块拥有低成本的优势。

NB-IoT 设备模块仅支持 FDD 半双工(Half-Duplex FDD，HD-FDD)模式，因此上、下行信号在频率上分开，模块将发送与接收分开处理，以达到节省元器件成本的目的。NB-IoT 模块在收发信号时，上行信号中前面的子帧和后面的子帧都只接收上行信号，不会接收到下行信号，从而延长保护时隙，降低设备需求，并且提高信号可靠性。此外，半双工设计中改变收、发模式只需增加切换器即可，比全双工成本更低；在简化设计的同时降低电池能耗。

因为运营商无须重新建设 NB-IoT 网络基站，所以除去运营商的成本外，NB-IoT 的成

本主要是成熟产业链带来的成本。现在市面上所有主流移动通信芯片和模块厂商都对 NB-IoT 计划给予大量支持，例如，华为、中兴、联发科、移远等制造商，这对于 NB-IoT 整个生态链都大有好处，并且还可大大降低制造成本。

4. NB-IoT 容量大

NB-IoT 的应用场景众多，例如，智慧农业、智能抄表等，这些应用主要以上传数据为主，下行数据相对较少，上行数据中子载波带宽有两种，分别是 15kHz 与 3.75kHz。NB-IoT 的上行数据业务主要由 MAR(Mobile Autonomous Reporting)例外报告和 MAR 周期性报告两部分组成。MAR 例外报告多应用于智能仪表的用电情况通知、意外报警通知等，此类应用场景属于事件报告类机制，设备模块只有在监测到某种特定事件发生后才发送上行数据，由于此类事件较为稀少(几个月或几年发生一次)，上行数据流量使用很低，因而此类应用场景对于 NB-IoT 网络的容量可以忽略不计。

5.1.2 NB-IoT 网络构架与部署模式

1. NB-IoT 网络构架

1) NB-IoT 主要网络构架

NB-IoT 作为新型物联网其众多优势特点源于 NB-IoT 网络构架的独特性，NB-IoT 的网络架构主要沿用的是 LTE 的网络架构，其网络架构主要由核心网与无线接入网(RAN)组成，如图 5-3 和图 5-4 所示。

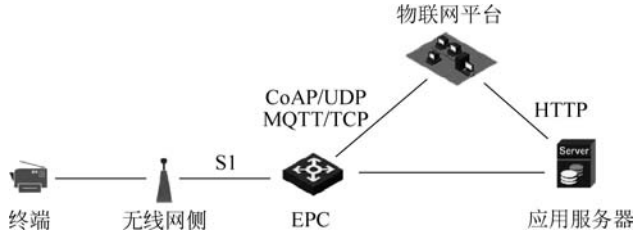

图 5-3 NB-IoT 主要网络构架

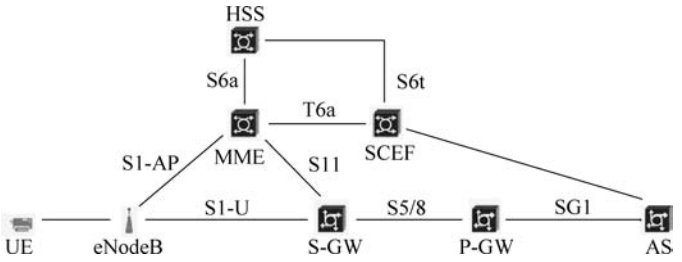

图 5-4 NB-IoT 网络结构框图

如图 5-3 所示，NB-IoT 主要网络构架由以下 5 个部分组成。

(1) 终端(User Equipment，UE)，通过空口接入连接到 LTE 架构基站 eNodeB(evolved NodeB，E-UTRAN 基站)。

(2) 无线网侧：无线网侧有整体式无线接入网(Singel RAN)与 NB-IoT 新建网这两种组网方式，整体式无线接入网包括 2G、3G、4G 和 NB-IoT 无线网，NB-IoT 新建网则主要用

S1-lite 接口与 IoT 核心网进行直接连接，将非接入层数据转发给高层网元处理，从而达到空口接入处理与网络小区管理等功能。

(3) EPC(Evolved Packet Core，演进数据封包核心网)：承担与 EU 非接入层交互的功能，并将数据信息转发到 IoT 平台进行数据处理。

(4) 物联网平台：现在国内物联网平台众多主要以 OneNET-中国移动物联网平台、中国电信物联网开放平台、中国联通物联网平台、阿里巴巴云物联网平台、华为 OceanConnect 物联网平台为主。

(5) 应用服务器：应用服务器使用内部通信协议和平台通信，通过调用物联网平台的开放端口来控制终端设备，物联网平台将终端设备上传的数据消息发送至应用服务器。

2) NB-IoT 网络结构细分

NB-IoT 网络主要有如下一些概念。

(1) MME(Mobility Management Entity，移动管理节点)是接入网络的关键控制节点，加入控制包括安全和许可控制，可在空闲模式下进行 UE 的寻呼控制。通过与 HSS(Home Subscribe Server，归属用户服务器)通信交流，完成用户安全许可验证。

(2) SCEF(Service Capa-bility Exposure Function，服务能力开放平台)为新增网元，支持对于新的 PDN(Public Data Network，公用数据网)类型 Non-IP 的控制侧数据传输。

(3) S-GW(Serving GW，服务网关)主要将用户数据包进行路由与转发。对于 UE 来说 S-GW 是下行数据的终点，并在下行数据到达时触发对 UE 的寻呼。

(4) P-GW(PDN GateWay，PDN 网关)，提供 UE 与外部分组数据网络连接点的接口传输，进行上、下行业务等级计费。

(5) X2 接口是在 eNodeB 之间进行消息命令传输和数据交互。NB-IoT 系统中新基站在 X2 接口向旧基站发起用户上下文获取流程，从旧基站获取终端在旧基站挂起时保存的用户上下文信息，以便在新基站上快速恢复该 UE。

(6) S1 接口分别作用于控制侧与用户侧两方面，以实现控制侧的 eNodeB 和 MME 之间消息命令的传输、用户侧的 eNodeB 和 SGW 之间的用户侧数据传输。在 NB-IoT 系统中，S1 接口特性主要包括无线接入技术数据上报、UE 无线传输能力消息指示、优化消息命令流程支持控制侧与用户侧优化数据传输方案等。

3) NB-IoT 传输方式

NB-IoT 包含 3 种传输方式：IP、Non-IP、SMS。

在扇区内 NB-IoT 的 UE 接入数量远大于 LTE 的 UE 接入数量(Long Term Evolution，LTE，意即长期演进，是由 3GPP 组织制定的通用移动通信系统技术标准的长期演进，于 2004 年 12 月在 3GPP 多伦多 TSG RAN＃26 会议上正式立项并启动)，则 NB-IoT 控制侧建立与释放的数据次数远大于 LTE；另外，LTE 的 UE 从空闲状态进入连接状态使用网络数据量较大。此外，LTE/EPC 的复杂指令流程对 UE 造成巨大能耗。从系统架构上进行分析，NB-IoT 从控制侧和用户侧方面对效率进行增强与优化。因此将 NB-IoT 传输优化方案分为控制侧传输优化方案与用户侧传输优化方案两种。

控制侧数据传输方式对小报数据传输进行优化，不用建立 DRB(Data Radio Bearer，无线承载)以及基站与 S-GW 间的 S1-U 承载。采用控制侧传输方案时，使用基站将小包数据通过 NAS 信令传输到 MME，并通过 MME 与 S-GW 之间建立连接，从而使小包数据在

MME 与 S-GW 之间进行传输。当 S-GW 收到下行传输数据时该连接依旧存在，S-GW 会将下行传输的数据发给 MME，反之则触发 MME 执行寻呼状态。控制侧传输方案有以下两个传输路径：

UE—eNodeB—MME—S-GW— P-GW

UE—eNodeB—MME—SCEF

用户侧传输是通过挂起流程与恢复流程，使用户从空闲状态迅速进入到连接状态。当 UE 从连接状态恢复到空闲状态时，MME 存储 UE 的 S1-AP 关联信息和承载上、下行数据，与此同时，不需要重新建立承载和安全信息的重协商。另外，小数据报文通过用户侧直接进行传输时，需要建立 S1-U 和 DRB。

2. NB-IoT 部署方式

在 NB-IoT 部署方式多样，在运营商未建设部署 LTE /FDD 网络的地区部署 NB-IoT，相当于重新建设网络成本极高，如果无现成空闲频段，还需要调整现有 GSM 网络频段。难度较高，如果有已部署的 LTE FDD 网络，部署 NB-IoT 可利用现成的 4G LTE 网络设备，通过部分软件升级进行部署。NB-IoT 可以直接部署在 GSM、UMTS(3GSM)或者 LTE 网络设施上，可以通过 LTR 网络平滑过渡，也可直接部署于单独的 180kHz 通信频段。NB-IoT 的 3 种部署方式为 ST(Stand-aloneoperation，独立部署)、GB(Guard Bandoperation，保护带部署)以及 IB(In-Bandoperation，带内部署)。

(1) 独立部署：不依靠 LTE 网络基站，可以重新利用 GSM 网络频段因为其带宽为 200kHz，所以 NB-IoT180kHz 带宽刚好可以有部署空间。独占部署不可与现有系统共存并属于频谱独占。NB-IoT 载波可以单独取代或者可以几个载波组合进行使用，多个载波捆绑在一起使用类似于 LTE 的 CA(Carrier Aggregation，载波聚合)技术，如图 5-5 所示。

(2) 保护带部署：利用现有 LTE 网络频谱外的带宽，也就是主频谱边缘保护频带中未使用的 180kHz 带宽。可以提升频谱资源利用率，如图 5-5 所示。

(3) 带内部署：需要运营商在搭建 LTE 网络时提前考虑带内部署 NB-IoT，使用多个 NB-IoT 载波来取代相应的多个 LTE 载波的 PRB(Physical Resource Block，物理资源块)，如图 5-5 所示。

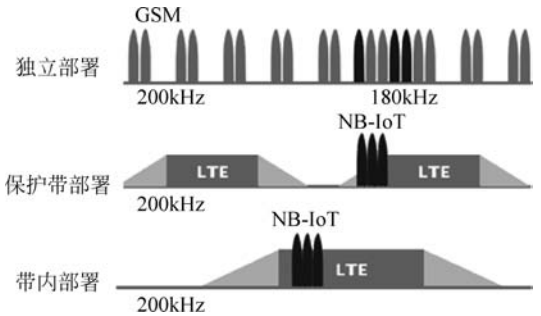

图 5-5 NB-IoT 的 3 种部署方式

NB-IoT 在 3GPP R13 中专门定义了半双工模式，该模式中 UE 无法同时进行接收与发送。半双工模式分为 Type A 和 Type B。

- Type A：UE 在上行信号发送时，前一子帧下行信号的最后一个符号不做接收，从

而作为 GP(Guard Period,保护时隙);

- Type B:UE 在上行信号发送时,前一个子帧与后一个子帧对下行信号都不做接收,从而增长 GP。

NB-IoT 支持 R13 中定义的半双工 Type B 模式,如图 5-6 所示。

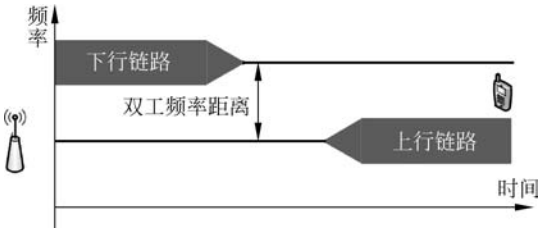

图 5-6 NB-IoT 网络半双工 Type B 模式

NB-IoT 只部署在 FDD LTE 网络上,FDD 就是上、下行频率分开,UE 也将发送与接收分开。半双工需要一个切换器,用来改变发送与接收模式切换,相比全双工,设计更简,成本更低。

3. NB-IoT 信道以及信道映射

1) NB-IoT 上行信道

NB-IoT 上行信道定义 NPUSCH(Narrowband Physical Uplink Shared Channel,窄带物理上行共享信道)和 NPRACH(Narrowband Physical Random Access Channel,窄带物理随机接入信道)两种物理信道。

NPUSCH 是发送上行消息与控制信息,使用多频音和单频音这两种模式。子载波 15kHz 间隔时连续包含 12 个子载波;子载波 3.75kHz 间隔时连续包含 48 个子载波。NPUSCH 以时域和频域两个维度资源的灵活组合为调度单位,不同于 LTE 系统中以 PRB 为调度单位,基本的调度单位为资源单位 RU,如表 5-2 所示。

表 5-2 物理上行共享信道对比

PUSCH	NB-IoT	Legacy LTE R8
频域	3.75kHz 间隔,单频音 15kHz 间隔,单频音 15kHz 间隔,12/6/3 tones	15kHz 子载波间隔,RB,12 个子载波
时域	15kHz 和 Legacy LTE 对齐 3.75kHz 下,定义 2ms Slot	1ms 子帧调度周期
信息	NPUSCH 1 上行数据 NPUSCH 2 ACK/NACK	PUSCH 上行数据,也可以携带 ACK/NACK
资源分配	按照 RU 分配资源 不同频域带宽对应不同 RU 资源时长	按照 RB 分配资源,RB 数量为 2、3、5 的倍数
编码	1/3 Turbo	1/3 Turbo
调制	Single Tone Pi/2-BPSK Single Tone Pi/4-QPSK Multi Tone QPSK	QPSK,16QAM
RV 版本	支持 RV0,RV2	支持 RV0,RV1,RV2,RV3

NPUSCH 支持两种格式。格式 1 承载上行共享信道，可以上传用户设备数据指令，通过一个或几个资源块进行调度，支持两种子载波传输方式；格式 2 承载上行控制信息，只能使用单频音模式。

2）NB-IoT 下行信道

NB-IoT 定义了 NPBCH（Narrowband Physical Broadcast Channel，窄带物理层广播信道）、NPDCCH（Narrowband Physical Downlink Control Channel，窄带物理层下行链路控制信道）和 NPDSCH（Narrowband Physical Downlink Shared Channel，窄带物理层下行链路共享信道）3 种不同的下行物理层信道。

NB-IoT 下行物理层信道都为 QPSK（Quadrature Phase Shift Keying，正交相移键控）；为减少用户 UE 编码译码难度，编码方式为 TBCC（Tail Biting Convolutional Coding，咬尾卷积码），降低了物联网系统的应用门槛。

NPBCH 一个广播周期 64 帧有 640ms，在 80ms 内传输一个数据块并且重复发送，数据块内包含 8 个独立解码数据块；NPBCH 每次传输都保持其位置在无线帧中的 0＃子帧上，每次传输 0＃子帧的前 3 个符号都不占用，如图 5-7 和图 5-8 所示。数据经 QPSK 调制解调后得出的编码，每个编码数据块都重复发送 8 次，因此每个无线帧上的 0 号子帧恰好都装满了 NPBCH 的符号。

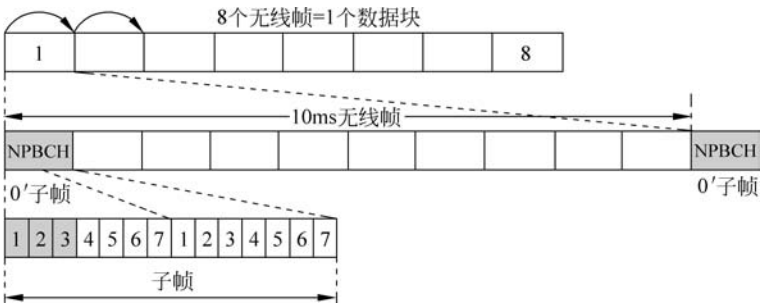

图 5-7 NB-IoT 窄带物理层广播信道

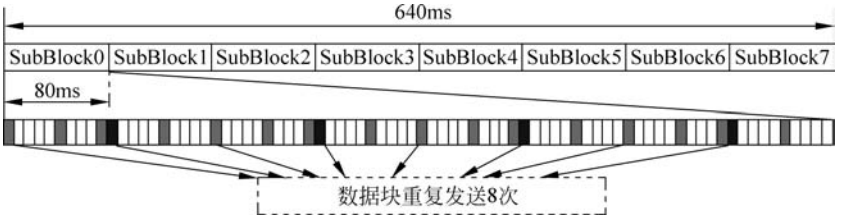

图 5-8 NPBCH 时频域映射

NPDCCH 承载多种 DCI（DownlinkControlInformation，下行控制信息），作为 NB-IoT 整个系统的控制核心，NPDCCH 中包含设备之间的资源分配和控制信息。NPDCCH 的信道处理有以下几个步骤：加扰处理、调制解调、层映射（Layer mapping）、预编码（Precoding）、资源映射等。加扰处理方法会重新初始化 NPDCCH 子帧的加扰序列；调制解调采用 QPSK 调制方式；层映射和预编码与 LTE 网络相似，使用与 NPBCH 相同的天线端口；在资源映射时需要注意避开广播信道、同步信号以及参考信号等信号频段。

5.2 LoRa 及 LoRaWAN

5.2.1 概述

LoRa 是低功耗广域网通信技术中的一种，是 Semtech 公司专有的一种基于扩频技术的超远距离无线传输技术。LoRaWAN 是为 LoRa 远距离通信网络设计的一套通信协议和系统架构。它是一种介质访问控制(MAC)层协议。LoRaWAN 在协议和网络架构的设计上，充分考虑了节点功耗、网络容量、QoS(Quality of Service，服务质量)、安全性和网络应用多样性等因素。使用 LoRaWAN 协议的网络可以完成安全的双向通信、移动化和本地服务，设置过程中的配置环节省时、省力、省事，因此物联网的使用者们(开发商、移动用户和使用单位)有更加自由的操作空间。

如果按协议分层 LoRaWAN 是 MAC 层(因此 LoRaWAN 原来也被叫作 LoRaMAC)，LoRa 是物理层。LoRa 与 LoRaWAN 的层次关系如图 5-9 所示。

图 5-9 LoRa 与 LoRaWAN 的层次关系

LoRaWAN 是一种流行且广泛部署的 LPWAN 通信标准，在 ISM(工业，科学，医疗)频段使用未经许可的无线电频谱，频率约为 900MHz 或 430MHz(世界各地的精确频率各不相同)。使用未经许可的频谱意味着公司可以轻松推出网络，并为企业提供专用网络。LoRaWAN 定义了网络的通信协议和系统架构。

LoRa 网络具有星状布局，其中数百或数千个设备与连接到核心网络的网关以及最终的互联网进行双向通信。来自单个传感器或设备的信号由范围内的所有网关接收，这提高了可靠性并开辟了定位服务的可能性。该网络使用复杂的“自适应数据速率”算法来微调每个设备和网关之间的通信，以最小化功耗并最大化可靠性。一个 LoRaWAN 架构中包含了终端、网关、服务器这 3 个部分，如图 5-10 所示。LoRa 节点一般与传感器连接，负责收集传感数据，然后通过 LoRaWAN 协议传输给网关。

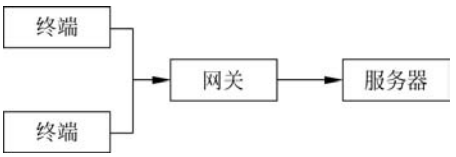

图 5-10 LoRaWAN 系统架构示意图

LoRaWAN 的优势主要包括：

1) 低功耗与使用寿命

低功耗和低峰值电流需求，LoRaWAN 数据传输和接收需要低电流(低于 50mA)，大大

降低了设备的功耗，一次充电可以使设备具有长达十年的使用寿命，大大降低了支持和维护成本。

2）节省成本

广泛的覆盖范围和相对较低的网关成本显著降低了 LoRaWAN 网络部署的成本。对于设备，通信模块的价格在 10 美元范围内，未经许可的频谱意味着连接成本仅为 1 美元/年。

3）定位服务

由于来自特定设备的信号可以被多个网关接收，因此可以根据每个基站的信号强度和/信号到达时间来计算设备的位置，从而实现基于位置的服务功能，可用于跟踪或对设备进行电子围栏。

4）深度穿透

LoRa 无线电调制允许深度室内穿透，并增加了到达位于地下的水表或煤气表的传感器的能力。

5）无须获取任何频率许可

LoRaWAN 网络部署在免费的 ISM 频段（EU 868、AS 923、US 915MHz）上，允许任何服务提供商或公司部署和运营 LoRaWAN 网络，而无须获得任何频率的许可证。

6）快速搭建和商用

LoRaWAN 开放标准与无成本运行频率和低成本基站相结合，使运营商能够在短短几个月内以最少的投资推出网络。双向通信完全支持各种需要上行链路和下行链路的用例，例如，街道照明、智能灌溉、能源优化或家庭自动化。

7）一站式管理

LoRaWAN 网络支持多种垂直解决方案，允许服务提供商使用一个平台和标准来管理各种用例，如智能建筑、精准农业、智能计量或智能城市。

5.2.2 工作模式和终端设备分类

在 LoRaWAN 协议中，终端有 3 种工作模式：双向终端设备（Class A）、具有确定接收时隙的双向终端设备（Class B）和具有最大化接收时隙的双向终端设备（Class C），3 种工作模式的特点见表 5-3。

表 5-3 LoRaWAN 3 种工作模式的特点

双向终端设备	具有确定接收时隙的双向终端设备	具有最大化接收时隙的双向终端设备
电池供应	低延时	最小延时
双向通信	在指定的持续时间内双向通信	双向通信
单播消息	单播和多播消息	单播和多播消息
终端设备启动通信（上行链路）	额外的接收窗口	服务器可以随时启动传输
服务器在预定的响应窗口期间与终端设备进行通信	服务器在指定的时隙内启动传输	终端设备不断接收

双向终端设备是 LoRa 网络中的基础工作模式，所有 LoRa 终端都必须满足双向终端设

备定义的所有功能。在双向终端设备工作模式下,节点主动上报,先发送一个上行链路帧,后紧跟两个下行接收窗口,要求应用在终端上行传输后的很短时间内进行服务器的下行数据传输,以此实现双向传输。双向终端设备模式是最省电的一种工作模式,终端只有在发送上行数据后,才可以接收网关下发的下行数据,服务器在其他任何时间进行的下行数据传输都需要等到终端的下一次上行数据上传之后。

在具有确定接收时隙的双向终端设备工作模式下,终端有更多的接收窗口,除了双向终端设备中两个随机时隙以外,终端设备还会在系统特定的持续时间内开启其余的接收窗口,持续时间由网关下发的时间同步信标帧指定。具有确定接收时隙的双向终端设备模式兼顾实时性和低功耗,对时间同步性要求较高。

具有最大化接收时隙的双向终端设备模式是常发常收模式,意味着接收窗口长时间处于开放状态,关闭的时段也仅仅是发送持续的时段,因此,具有最大化接收时隙的双向终端设备模式下的 LoRa 终端设备比双向终端设备和具有确定接收时隙的双向终端设备对应的设备消耗更多电量,但是设备的延时也最低,实时性最好,适合不考虑功率或需要大量下行数据控制的应用场景。

网关是部署 LoRaWAN 网络的关键设备,主要负责将终端节点采集的传感器数据传输给服务器,缓解海量节点数据上报引发的并发冲突,同时完成数据从 LoRa 方式到网络方式的转换。其中,网关收到上行的传感器数据后,不对数据包进行解析和解密处理,只封装成 JSON(JavaScript Object Notation)数据包然后上传至 LoRa 服务器。LoRaWAN 网关的主要特点如下。

(1) 网络拓扑简单,星状网络可靠性高,功耗低;

(2) 接入方式可扩展,单网关最大能够容纳几万个节点,节点非固定入网,数目可扩展;

(3) 兼容性强,只要应用能够符合 LoRaWAN 协议,该应用就是可接入的;

(4) 网络建设成本和运营成本较低。

根据 LoRaWAN 的规定,LoRa 服务器保证了 LoRaWAN 系统的运行维护和数据处理过程,并向下发送主要的控制命令。LoRa 服务器包括 3 部分,分别是网络服务器(Network Server,NS)、应用服务器(Application Server,AS)和客户服务器(Customer Server,CS),每个部分的分工和职能都各不相同,其中 NS 和 AS 是完成 LoRaWAN 协议必不可少的重要组成部分。其中与网关进行通信的是 NS,NS 收到网关发来的信息后,首先验证数据的合法性,验证通过后从中提取数据,对数据进行解析,然后整理成 NS 的 JSON 数据包上传至 AS。数据包上传至 AS 后,AS 负责上行数据的解密和下行数据的加密,以及对终端节点的入网请求进行处理,并生成两个重要的通信密钥——NwkSKey 和 AppSKey,NwkSKey 用于数据的校验,AppSKey 用于数据的加密和解密。客户服务器负责将 AS 发送的数据处理成用户自定义的数据协议格式,主要的表现形式为使用者开发的基于 B/S 或 C/S 架构的服务器,用于处理实际应用过程中的业务和数据展示部分。

LoRaWAN 终端设备根据协议规定被分为 Class A/Class B/Class C 3 类终端设备,这 3 类设备基本覆盖了物联网所有的应用场景。Class A/Class B/Class C 的应用和区别如表 5-4 所示。

表 5-4 Class A/Class B/Class C 的应用和区别

Class	介绍	下行时机	应用场景
A(全部)	所有 LoRaWAN 设备必须实现 Class A 的功能,在节点每次发送上行数据后都会紧跟两个短暂的下行接收窗口,以此实现双向通信,Class A 主打低功耗	必须等待终端节点上报数据后才能下发数据	主要用于小数据采集控制实时性不高的场景,如垃圾桶检测、烟雾报警、气体监测等
B(信标)	除了 Class A 的接收窗口,还会在指定时间打开接收窗口。为了让终端可以在指定时间打开接收窗口,终端需要从网关接收同步时间的信标	在约定的时间下发数据,相比于 Class A 控制的实时性有所提高	阀控水、汽、电表等
C(持续)	Class C 和 Class A 比较类似,只是在 Class A 休眠的时间打开接收窗口 2,Class C 是没有休眠的,除了发送的时候,其他时间要么开着接收窗口 1,要么开着接收窗口 2	任意时刻	路灯控制等

5.2.3 LoRaWAN 帧结构

上行链路信息是由传感器向上发送,在传送过程中跳转经过一个或者多个网关到达网络服务器的传感器数据信息。上行链路的物理帧结构如图 5-11 所示。下行链路信息则是由网络服务器发送给终端设备的信息,每条信息对应的监测设备都是唯一确定的,并且只经过一个网关中转。下行链路的物理帧结构如图 5-12 所示。

图 5-11 上行链路物理帧结构

图 5-12 下行链路物理帧结构

上下行物理帧中都包含前导码(Preamble)、物理帧头(PHDR)、循环冗余校验的物理帧头(PHDR_CRC)和有效负载(PHYPayload),不同的是,上行链路物理帧的最后还有循环冗余校验(CRC),下行链路物理帧中没有,CRC 的作用是保证有效负载的完整性。物理帧中 Preamble、PHDR、PHDR_CRC 和 CRC 都是由硬件设备生成的,无须软件参与,只有有效负载 PHYPayload 需要软件的参与。

以上行链路信息为例,其 MAC 信息帧的结构如图 5-13 所示。

PHYPayload 以单字节的 MAC 层帧头(MHDR)开始,以 4B 的消息一致码(MIC)结尾,其余部分是 MAC 层负载(MACPayload)。MHDR 中有区别消息类型的 MType 字段。LoRaWAN 定义了 6 种不同的 MAC 消息类型,不同的消息类型用不同的方法保证消息的一致性。MAC 层负载又由 MAC 层负载头(FHDR)、MAC 层数据的通道号(FPORT)和加密的 MAC 层负载(FRMPayload)3 部分组成。FHDR 由 4B 的终端地址(DevAddr)、1B 的

图 5-13 MAC 信息帧的结构

帧控制字(FCtrl)、2B 的帧序号(FCnt)和用来传输 MAC 命令的帧配置字段(FOpts)4 个部分组成。

5.2.4 LoRaWAN 网络架构和入网模式

LoRaWAN 的网络架构如图 5-14 所示。LoRaWAN 的网络实体分为 4 个部分：终端节点、LoRa 网关、LoRaWAN 服务器和应用服务器。

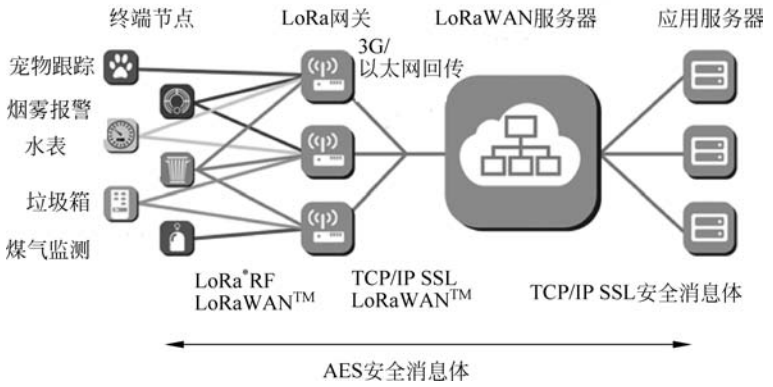

图 5-14 LoRaWAN 的网络架构

- 终端节点(End Node)：一般是各类传感器，进行数据采集、开关控制等。
- LoRa 网关(Gateway)：对收集到的节点数据进行封装转发。
- LoRaWAN 服务器(NetworkServer)：主要负责上下行数据包的完整性校验。
- 应用服务器(ApplicationServer，AS)：主要负责 OTAA 设备的入网激活，应用数据的加密和解密。用户服务器 customerServer 从 AS 中接收来自节点的数据，进行业务逻辑处理，通过 AS 提供的 API 接口向节点发送数据。

LoRa 终端节点在上电之后处于非入网模式，需要先激活入网才能和服务器进行通信，LoRaWAN 主要的入网模式分为两种：一种模式是手动激活(Activation By Personalization，ABP)，另一种模式是空中激活(Over-The-Air Activation，OTAA)。接下来对两种入网模式进行介绍。

OTAA 是一种安全系数很高的入网机制，它的代价是实现复杂；ABP 是一种简单的入网机制，适合于建设私网，缺点是不如 OTAA 模式安全。无论是哪种激活方式，其核心原理

都是在服务器和终端双方保存以下 3 个参数：DevAddr(终端设备地址)、NwkSKey(网络会话密钥)和 AppSKey(应用会话密钥)。DevAddr 是终端在当前网络中的识别码，该地址可以由网络管理员分配；NwkSKey 是分配给终端设备的网络会话密钥，服务器和终端用它来计算和校验所有信息的 MIC，保证收发数据的一致，也可以对 MAC 负载进行加/解密；AppSKey 是分配给终端设备的应用会话密钥，服务器和终端设备用来对应用指定的负载字段进行加密和解密，也可以用来计算和校验应用层 MIC。

ABP 模式时，直接把以上 3 个参数存到终端设备中，终端设备启动时就已经具备了加入指定 LoRa 网络所需要的信息。OTAA 模式的入网原理如图 5-15 所示。

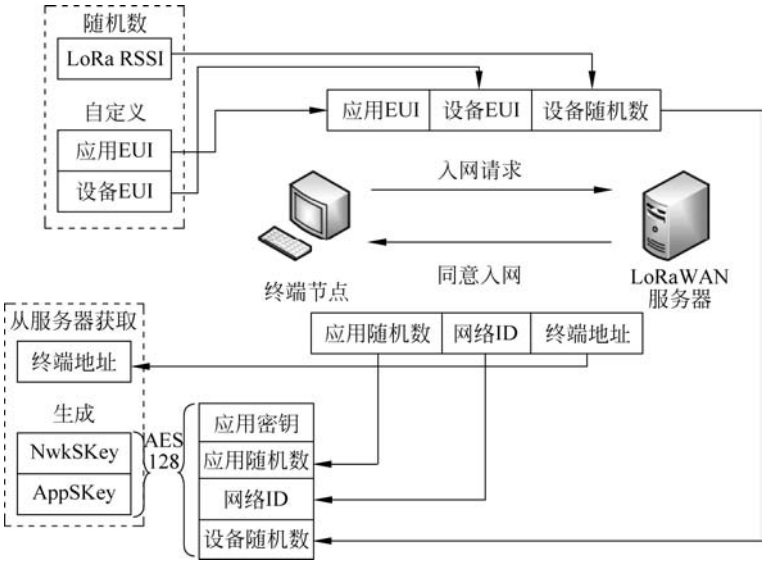

图 5-15　OTAA 模式的入网原理图

首先，每个终端设备都需要配置设备 EUI(Device Extended Unique Identifier，设备扩展唯一标识)和应用 EUI(Application Extended Unique Identifier，应用扩展唯一标识)，并且取 LoRa 芯片的 RSSI 随机值，得到设备随机数(Dev Nonce)，将这 3 个参数组成入网请求数据帧，发送给 LoRaWAN 服务器。服务器接收到入网请求帧后，分配 DevAddr，连同应用随机数(AppNonce)和网络 ID 组成统一入网数据帧，回应给终端设备。终端设备收到同意入网数据帧后，提取 DevAddr，结合应用密钥(AppKey)(被服务器和设备终端共享使用的重要数据)、AppNonce、网络 ID 和 DevNonce，使用 AES128 加密生成两个密钥：NwkSKey 和 AppSKey，完成入网。

最后从以下几个方面对 NB-IoT 技术和 LoRa 技术进行对比。

1) 通信距离

NB-IoT 和 LoRa 都属于低功耗广域网技术，即远程覆盖是它们的共同特点之一。NB-IoT 的通信距离可达 18～21km，不过 NB-IoT 的通信依赖运营商的蜂窝数据，所以在郊区或农村等没有强大 4G 覆盖的地区，通信效果稍逊于城市等通信状况较好的地区。LoRa 技术在郊区无线通信距离最高可达 20km，且 LoRaWAN 不依赖蜂窝数据或 WiFi，在 LoRaWAN 能够覆盖的区域内通信效果足够稳健。

2) 电池寿命

LoRaWAN 是基于 ALOHA 协议的异步通信方式,可以通过实际的使用需求进行工作状态的调节,继而做到准确的休眠时间设置,这样可以达到充分利用电池电量的目的。NB-IoT 工作在 1GHz 以下的授权频段,设备必须相对频繁地与网络进行同步,同步的过程需要联网,这样所需要的"峰值电流"比采用非线性调制的 LoRa 技术多几个数量级,使得电池电量有较大的压力。

3) 组网成本

对终端节点而言,LoRa 比 NB-IoT 的终端节点更加简单且容易开发,NB-IoT 的协议和调制机制比较复杂,那么在电路的设计上需要花费更多的工夫,也就意味着更高的成本,同时 NB-IoT 采用授权频段,知识产权相关费用更高,提高了 NB-IoT 的总成本。同时,由于 LoRa 工作在非授权频段,企业可以轻松组建私网,并同时使用公共网络处理设备外信息和活动,创建混合型物联网模型,而 NB-IoT 只能用于公共网络模式。

5.3 MQTT 协议

5.3.1 概述

MQTT(Message Queuing Telemetry Transport,消息队列遥测传输协议)是一种基于发布/订阅(publish/subscribe)模式的轻量级通信协议,该协议构建于 TCP/IP 协议上,由 IBM 在 1999 年发布。

MQTT 是一个基于客户端-服务器的消息发布/订阅传输协议。MQTT 协议是轻量、简单、开放和易于实现的,这些特点使其适用范围非常广泛。比如在机器与机器(M2M)通信领域和物联网领域这些受限的环境中。另外,MQTT 协议在通过卫星链路通信的传感器、智能家居、智慧城市等环境中也广泛使用。

OASIS(结构化信息标准促进组织)现在已经发布了官方的 MQTT v5.0 标准,这对于已经为物联网所用的消息传输协议来说是一个大飞跃。基于早期的 v3.1.1 标准,它具有重要的更新,同时最大限度地减少与现有版本的不兼容性。除标准版外,还有一个简化版 MQTT-SN,该协议主要针对嵌入式设备,这些设备一般工作于 TCP/IP 网络,如 ZigBee。

由于物联网的环境是非常特别的,所以 MQTT 遵循以下设计原则:

(1) 精简,不添加可有可无的功能。

(2) 发布/订阅(Pub/Sub)模式,方便消息在传感器之间传递。

(3) 允许用户动态创建主题,零运维成本。

(4) 把传输量降到最低以提高传输效率。

(5) 把低带宽、高延迟、不稳定的网络等因素考虑在内。

(6) 支持连续的会话控制。

(7) 客户端计算能力可能很差。

(8) 提供服务质量管理。

(9) 假设数据不可知,不强求传输数据的类型与格式,保持灵活性。

与 HTTP、CoAP、XMPP 等通信协议相比,MQTT 协议具有以下的优势:

(1) MQTT 基于 TCP 的协议,在反控设备的应用中比 CoAP 等基于 UDP 的协议更为

可靠，比如 CoAP 在进行 3G 通信的时候必须通过相关技术实现 CoAP overTCP，否则反控很不稳定。

(2) MQTT 通过异步模式实现通信，类似发短信，无须等待对方确认便可以继续发送信息，而不像 HTTP 那种必须等待对方应答才能返回的同步模式，大大简化了连接过程。

(3) MQTT 为物联网通信提供了许多满足特定需求的设计，比如 QoS、“遗嘱(用于通知同一主题下的其他设备发送遗言的设备已经断开了连接)”等。

(4) MQTT 的二进制格式是轻量级的，几乎能轻易嵌入到任何终端中，终端通信模块的功耗也大大降低，尤其适合无线终端的电池供电工作模式。

轻巧可扩展、对低功耗低速率网络的绝佳适应性，再加上 MQTT 完全开源开放，国内外的公有云供应商如阿里巴巴、百度、腾讯、AWS、Microsoft Azure、IBM Bluemix 等都以各种形式加入了对 MQTT 的支持，已经形成了良好的 MQTT 生态圈。因此，MQTT 协议是当前符合上述要求的物联网通信协议。

MQTT 协议的主要实现和应用包括：

(1) 已经有 PHP、Java、Python、C、C# 等多个语言版本的协议框架。

(2) IBM Bluemix 的一个重要部分是其 IoT Foundation 服务，这是一项基于云的 MQTT 实例。

(3) 移动应用程序也早就开始使用 MQTT，如 Facebook Messenger 和 Com 等。

5.3.2 MQTT 协议工作原理

1. 工作过程

MQTT 的工作原理如下：

(1) 使用发布/订阅消息的模式，提供一对多的消息发布，并实现应用程序的解耦合。

(2) 实现对负载内容进行屏蔽的消息传输。

(3) 使用 TCP/IP 提供网络连接，相比 HTTP 协议，MQTT 数据包的头部开销非常小，头部是固定的，且只有两字节。

(4) 提供 3 种消息发布的服务，用户能够针对网络的实际情况以及服务要求选择 3 种不同的通信质量等级，分别是：

① 至多一次，消息发布完全依赖底层网络的 TCP/IP，可能发生消息丢失。这一等级适用于环境数据采集等应用场景，丢失一次数据影响不大，因为不久后还会有第二次的数据发送。

② 至少一次，能够确保消息到达，但消息可能会重复发送。

③ 只有一次，确保消息到达一次。这一等级适用于可靠的消息送达的应用场景，例如计费系统。

(5) 支持遗嘱机制。根据用户设置的遗嘱机制，当由于网络因素等非正常原因导致终端连接断开时，将以发布话题的方式通知可能对该终端感兴趣的其他终端。

MQTT 协议中主要包含了消息发布者(Publish)、消息订阅者(Subscribe)以及消息代理(Broker)。消息发布者和消息订阅者的主要工作是发布消息和订阅消息，所以两者是以客户端的形式存在，客户端可以：

(1) 发布其他客户端可能会订阅的信息。

(2) 订阅其他客户端发布的消息。

(3) 退订或删除应用程序的消息。

(4) 断开与服务器连接。

消息代理作为消息的“中转商”,所以是以服务器的形式存在,MQTT 服务器可以:

(1) 接收来自客户的网络连接。

(2) 接收客户发布的应用信息。

(3) 处理来自客户端的订阅和退订请求。

(4) 向订阅的客户转发应用程序消息。

消息的发布/订阅过程作为一个双向通信的过程,消息发布者和消息订阅者两者之间可以相互订阅,两者的身份可以互相转换。MQTT 协议原理如图 5-16 所示。

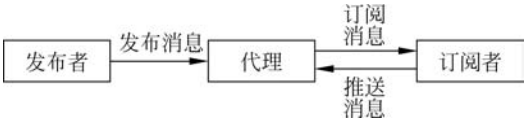

图 5-16 MQTT 协议原理图

MQTT 传输的消息分为主题(Topic)和负载(payload)两部分。

(1) 主题,可以理解为消息的类型,订阅者订阅(Subscribe)后,就会收到该主题的消息内容(payload)。

(2) 负载,可以理解为消息的内容,是指订阅者具体要使用的内容。

2. MQTT 协议中的订阅、主题、会话

1) 订阅(Subscription)

订阅包含主题筛选器(Topic Filter)和最大服务质量(QoS)。订阅会与一个会话(Session)关联。一个会话可以包含多个订阅。每一个会话中的每个订阅都有一个不同的主题筛选器。

2) 会话(Session)

每个客户端与服务器建立连接后都是一个会话,客户端和服务器之间有状态交互。会话存在于一个网络之间,也可能在客户端和服务器之间跨越多个连续的网络连接。

3) 主题名(Topic Name)

连接到一个应用程序消息的标签,该标签与服务器的订阅相匹配。服务器会将消息发送给订阅所匹配标签的每个客户端。

4) 主题筛选器(Topic Filter)

一个对主题名通配符筛选器,在订阅表达式中使用,表示订阅所匹配到的多个主题。

5) 负载(Payload)

消息订阅者所具体接收的内容。

3. MQTT 协议中的方法

MQTT 协议中定义了一些方法(也被称为动作)来表示对确定资源所进行操作。这个资源可以代表预先存在的数据或动态生成数据,这取决于服务器的实现。通常来说,资源指服务器上的文件或输出。主要方法有:

(1) Connect。等待与服务器建立连接。

(2) Disconnect。等待 MQTT 客户端完成所做的工作,并与服务器断开 TCP/IP 会话。

(3) Subscribe。等待完成订阅。

(4) UnSubscribe。等待服务器取消客户端的一个或多个主题订阅。

(5) Publish。MQTT 客户端发送消息请求,发送完成后返回应用程序线程。

5.3.3 MQTT 协议数据包结构

在 MQTT 协议中,一个 MQTT 数据包由固定头(Fixed header)、可变头(Variable header)、消息体(Payload)3 部分构成。MQTT 数据包结构如图 5-17 所示。

图 5-17 MQTT 数据包结构

(1) 固定头存在于所有 MQTT 数据包中,表示数据包类型及数据包的分组类标识。

(2) 可变头存在于部分 MQTT 数据包中,数据包类型决定了可变头是否存在及其具体内容。

(3) 消息体存在于部分 MQTT 数据包中,表示客户端收到的具体内容。

1. MQTT 固定头

固定头存在于所有 MQTT 数据包中,其结构如图 5-18 所示。

地址 \ bit	7	6	5	4	3	2	1	0
Byte 1	MQTT数据包类型				不同类型MQTT数据包的具体标识			
Byte 2 ⋮	剩余长度							

图 5-18 MQTT 固定头

如图 5-18 所示,MQTT 数据包类型位于 Byte 1 中的 bit7～bit4。相于一个 4 位的无符号值,类型、取值及描述如表 5-5 所示。

表 5-5 MQTT 数据包类型

名称	值	流方向	描述
Reserved	0	不可用	保留位
CONNECT	1	客户端到服务器	客户端请求连接到服务器
CONNACK	2	服务器到客户端	连接确认
PUBLISH	3	双向	发布消息
PUBACK	4	双向	发布消息
PUBREC	5	双向	发布收到(保证第 1 部分到达)
PUBREL	6	双向	发布释放(保证第 2 部分到达)
PUBCOMP	7	双向	发布完成(保证第 3 部分到达)
SUBSCRIBE	8	客户端到服务器	客户端请求订阅
SUBACK	9	服务器到客户端	订阅确认
UNSUBSCRIBE	10	客户端到服务器	请求取消订阅

续表

名称	值	流方向	描述
UNSUBACK	11	服务器到客户端	取消订阅确认
PINGREQ	12	客户端到服务器	PING 请求
PINGRESP	13	服务器到客户端	PING 应答
DISCONNECT	14	客户端到服务器	中断连接
Reserved	15	不可用	保留位

标识位位于 Byte 1 中的 bit 3～bit 0。在不使用标识位的消息类型中,标识位被作为保留位。如果收到无效的标志,接收端必须关闭网络连接。表 5-6 列举了标识位在不同情况下的取值。

表 5-6 标识位的取值

数据包	标识位	bit 3	bit 2	bit 1	bit 0
CONNECT	保留位	0	0	0	0
CONNACK	保留位	0	0	0	0
PUBLISH	MQTT 3.1.1 使用	DUP[1]	QoS[2]	QoS[2]	RETAIN[3]
PUBACK	保留位	0	0	0	0
PUBREC	保留位	0	0	0	0
PUBREL	保留位	0	0	0	0
PUBCOMP	保留位	0	0	0	0
SUBSCRIBE	保留位	0	0	0	0
SUBACK	保留位	0	0	0	0
UNSUBSCRIBE	保留位	0	0	0	0
UNSUBACK	保留位	0	0	0	0
PINGREQ	保留位	0	0	0	0
PINGRESP	保留位	0	0	0	0
DISCONNECT	保留位	0	0	0	0

注:1. DUP:发布消息的副本。用来保证消息的可靠传输,如果设置为 1,则在下面的变长中增加 MessageId,并且需要回复确认,以保证消息传输完成,但不能用于检测消息重复发送。

2. QoS:发布消息的服务质量,即保证消息传递的次数。

- 00:最多一次,即≤1。
- 01:至少一次,即≥1。
- 10:一次,即=1。
- 11:预留。

3. RETAIN:发布保留标识,表示服务器要保留这次推送的信息,如果有新的订阅者出现,则将消息推送给它;如果没有,则推送至当前订阅者后释放。

剩余长度(Remaining Length)的地址是 Byte 2。

固定头的第二字节用来保存变长头部和消息体的总大小,但不是直接保存的。这一字节是可以扩展的,其前 7 位用于保存长度,最后 1 位用作标识。当最后 1 位为 1 时,表示长度不足,需要使用两字节继续保存。

2. MQTT 可变头

MQTT 数据包中包含一个可变头，它位于固定的头和负载之间。可变头的内容因数据包类型而不同，较常见的应用是作为包的标识，表 5-7 列举了可变头的取值情况。

表 5-7 可变头的取值

长度/B	位								存在的报文类型
	7	6	5	4	3	2	1	0	
1～8	Protocol name(协议名)								CONNECT
1	Protocol Version(版本号)								CONNECT
1	Connect Flags(连接标记)								CONNECT
	User Name Flag	Password Flag	Will Retain	Will QoS		Will Flag	Clean Session	Reserved	CONNECT
2	Keeping Alive timer(心跳时长)								CONNECT
1	Connect return code(连接返回码)								CONNACK
3～32 767	Topic name(主题名称)								PUBLISH
2	Message ID(消息 ID)								PUBLISH(QoS＞0)，PUBACK，PUBREC，PUBREL，PUBCOMP，SUBSCRIBE，SUBACK，UNSUBSCRIBE，UNSUBACK

很多类型数据包中都包括一个 2B 的数据包标识字段，这些类型的包有 PUBLISH(QoS＞0)、PUBACK、PUBREC、PUBREL、PUBCOMP、SUBSCRIBE、SUBACK、UNSUBSCRIBE、UNSUBACK。

3. Payload 消息体

Payload 消息体位于 MQTT 数据包的第三部分，包含 CONNECT、SUBSCRIBE、SUBACK、UNSUBSCRIBE 4 种类型的消息。

(1) CONNECT：消息体内容主要是客户端的 ClientID、订阅的 Topic、Message 以及用户名和密码。

(2) SUBSCRIBE：消息体内容是一系列的要订阅的主题以及 QoS。

(3) SUBACK：消息体内容是服务器对于 SUBSCRIBE 所申请的主题及 QoS 进行确认和回复。

(4) UNSUBSCRIBE：消息体内容是要订阅的主题。

5.4 CoAP 协议

CoAP(Constrained Application Protocol，受限应用协议)，是应用于无线传感网中的协议。由于目前物联网中的很多设备都是资源受限型的，只有少量的内存空间和有限的计算

能力,传统的 HTTP 协议在物联网应用中就会显得过于庞大而不适用,因此,IETF 的 CoRE 工作组提出了一种基于 REST 架构、传输层为 UDP、网络层为 6LowPAN(面向低功耗无线局域网的 IPv6)的 CoAP 协议。

CoAP 协议具有如下特点。

- 报头压缩:CoAP 包含一个紧凑的二进制报头和扩展报头。它只有短短的 4B 基本报头,基本报头后面跟扩展选项。一个典型的请求报头为 10～20B。
- 方法和 URI:为了实现客户端访问服务器上的资源,CoAP 支持 GET、PUT、POST 和 DELETE 等方法。CoAP 还支持 URI(Uniform Resource Identifier,统一资源标识符),这是 Web 架构的主要特点。
- 传输层使用 UDP 协议:CoAP 协议建立在 UDP 协议之上,以减少开销和支持组播功能。它也支持一个简单的停止和等待的可靠性传输机制。
- 支持异步通信:HTTP 对 M2M(Machine-to-Machine)通信不适用,这是由于事务总是由客户端发起。CoAP 协议支持异步通信,这对 M2M 通信应用来说是常见的休眠/唤醒机制。
- 支持资源发现:为了自主地发现和使用资源,CoAP 支持内置的资源发现格式,用于发现设备上的资源列表,或者用于设备向服务目录公告自己的资源。它支持 RFC5785 中的格式,在 CoRE 中用/.well-known/core 的路径表示资源描述。
- 支持缓存:CoAP 协议支持资源描述的缓存以优化其性能。

CoAP 默认运行在 UDP 上,但它也支持运行在 SMS、TCP 等数据传输层上。本节主要介绍基于 UDP 上的 CoAP 协议。CoAP 协议框架如图 5-19 所示。CoAP 协议主要包括消息模型和资源请求/响应模型。

图 5-19 CoAP 协议框架

1. 消息模型

CoAP 协议通信是通过在 UDP 上传输消息类完成的。若将 UDP 比作公路,消息就是公路上的汽车。

CoAP 采用与 HTTP 协议相同的请求响应工作模式,CoAP 协议共有 4 种不同的消息类型,来实现设备端与云端之间双向通信。

- CON——需要被确认的请求,如果 CON 请求被发送,那么对方必须做出响应。
- NON——不需要被确认的请求,如果 NON 请求被发送,那么对方不必做出回应。
- ACK——应答消息,接收到 CON 消息的响应。
- RST——复位消息,若接收者接收到的消息包含一个错误,接收者解析消息或者不再关心发送者发送的内容,那么复位消息将会被发送。

基于 4 种消息类型,可以实现 2 种传输质量,即可靠消息传输与不可靠消息传输。

可靠消息传输主要是通过确认及重传机制来实现的,客户端发送消息后,需要等待服务

器收到通知，如果在规定时间内没有收到消息，则需要重新发送数据。可靠传输是基于CON消息传输的，服务器端收到CON类型的消息后，需要返回ACK消息，客户端在指定时间ACK_TIMEOUT内收到ACK消息后，才代表这个消息可以可靠地传输到服务器端。

不可靠消息传输是客户端只管发送消息，不管服务器端有没有收到，因此可能存在丢包。不可靠传输是基于NON消息传输的。服务器端收到NON类型的消息后，不用回复ACK消息。

2. 资源请求/响应模型

对于物联网，可以将服务器上的资源简单看作为物联网设备的实时运行影子，通过访问服务器上资源就可以实现与设备间数据的交互。如果把消息模型比作汽车，那么资源请求及响应就好比汽车上的货物。资源请求及响应内容最终会被放在CoAP消息包里面。CoAP请求与响应与HTTP类似，且是根据RESTFUL架构设计的。CoAP客户端发出请求，CoAP服务端进行请求处理然后发送响应。

CoAP请求Request方法：请求方法与HTTP协议类似，有GET、PUT、POST、DELETE，所有的请求方法都会放在CoAP CON/NON消息里面进行传输。

CoAP请求响应Response代码：响应内容也与HTTP协议类似。主要有如下3类：

- Success 2.xx代表客户端请求被成功接收并被成功处理；
- Client Error 4.xx代表客户端请求有错误，比如参数错误等；
- Server Error 5.xx代表服务器在执行客户端请求时出错。

所有的请求服务器响应放在CoAP CON/NON/ACK消息里面进行传输。针对CoAP携带CON消息请求，响应如果快速处理完（有些请求的处理耗时多，服务器无法立即响应），则可直接放在ACK消息包里面返回。对于无法立即响应的，服务器携带资源准备好后，会单独发一个响应消息包给客户端

服务器上可访问资源统一用URL来定位（比如/deviceID/temp访问某个设备的温度信息）。客户端通过某个资源的URL来访问服务器具体资源，通过4个请求方法（GET、PUT、POST、DELETE）完成对服务器上资源的增、删、改、查操作。

举个例子，比如某个设备需要从服务器端查询当前温度信息。

请求消息（CON）：GET/temperature，请求内容会被包在CON消息里面。

响应消息（ACK）：2.05 Content “22.5 C”，响应内容会被放在ACK消息里面。

通信过程如图5-20所示。

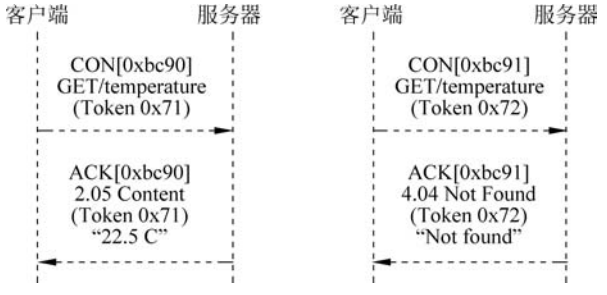

图5-20 通信过程

MQTT 和 CoAP 都是行之有效的物联网协议,但两者有很大区别,比如 MQTT 协议是基于 TCP,而 CoAP 协议是基于 UDP 的。从应用方向来分析,主要区别有以下几点:

(1) MQTT 协议不支持带有类型或者其他帮助客户端理解的标签信息,也就是说,所有 MQTT Clients 必须要知道消息格式。而 CoAP 协议则相反,因为 CoAP 内置发现支持和内容协商,这样便允许设备相互窥测以找到数据交换的方式。

(2) MQTT 是长连接而 CoAP 是无连接。MQTT 客户端与代理之间保持 TCP 长连接,这种情形在 NAT 环境中不会产生问题。如果在 NAT 环境下使用 CoAP,则需要采取一些 NAT 穿透性手段。

(3) MQTT 是多个客户端通过中央代理进行消息传递的多对多协议。它主要通过让客户端发布消息、代理决定消息路由和复制来解耦消费者和生产者。MQTT 相当于消息传递的实时通信总线。CoAP 基本上就是一个在服务器和客户端之间传递状态信息的单对单协议。

表 5-8 列举了 MQTT 与 CoAP 的对比情况。

表 5-8 MQTT 与 CoAP 的对比

项目	MQTT	CoAP
通信机制	异步	同步
连接方式	TCP	UDP
通信模式	多对多(服务器对服务器,设备对服务器,设备对 App)	多(设备)对一(服务器,系统架构类似于传统 Web)
使用场景	更适用于推送和 IM	物联网
功耗	功耗高	功耗低
支持平台	华为、电信、移动	阿里巴巴云、百度、腾讯 QQ 物联平台、中移动 OneNet、Amazon IoT 服务
反向控制	可反向控制	CoAP 不能反向控制

5.5 LwM2M 协议

5.5.1 概述

LwM2M 全称 lightweight Machine to Machine,是 OMA(Open Mobile Alliance,开放移动通信联盟)定义的物联网协议,主要使用在资源受限(包括存储、功耗等)的嵌入式设备上。

LwM2M 协议支持多 LwM2M Server 和基于资源模型的简单对象,支持 TLV/JSON/Plain Text/Opaque 等数据格式,使用 DTLS 安全协议,可以对资源实现创建/检索/更新/删除/属性配置等操作。

LwM2M 定义了两个逻辑实体:LwM2M 服务器和 LwM2M 客户端,客户端负责执行服务器的命令和上报执行结果。

在这两个逻辑实体之间有 4 个逻辑接口:

(1) 设备发现和注册(Device Discovery and Registration)。

该接口让客户端注册到服务器并通知服务器客户端所支持的能力(也就是说,支持哪些资源 Resource 和对象 Object)。

(2) 引导程序(Bootstrap)。

Bootstrap server 通过该接口来配置客户端,比如 LwM2M 服务器的 URL 地址。

(3) 设备管理和服务实现(Device Management and Service Enablement)。

该接口是最主要的业务接口。LwM2M 服务器发送指令给客户端并收到回应。

(4) 信息上报(Information Reporting)。

LwM2M 客户端利用该接口上报其资源信息,比如传感器温度。上报方式可以是事件触发,也可以是周期性的。

这些接口的操作目标为对象(Object)、对象实例(Object Instance)及资源(Resources)。

对象是逻辑上用于特定目的的一组资源的集合。例如,固件更新对象,它就包含了用于固件更新目的的所有资源,例如,固件包、固件 URL、执行更新、更新结果等。使用对象的功能之前,必须对该对象进行实例化,对象可以有多个对象实例,对象实例的编号从 0 开始递增。

OMA 定义了一些标准对象,LwM2M 协议为这些对象及其资源已经定义了固定的 ID。为实现不同功能划分,协议定义了 8 种对象,其 ID 及名称分别为 0. Security Object、1. Server Object、2. Access Control Object、3. Device Object、4. Connectivity Monitoring Object、5. Firmware Update Object、6. Location Object、7. Connectivity Statistics Object。例如,固件更新对象的对象 ID 为 5,该对象内部有 8 个资源,资源 ID 分别为 0～7,其中"固件包名字"这个资源的 ID 为 6。因此,URI 5/0/6 表示:固件更新对象第 0 个实例的固件包名字这个资源。

具体标准对象如表 5-9 所示。

表 5-9 OMA 定义的标准对象

对 象	对象 ID	说 明
LwM2M 安全对象(LwM2M Security)	0	LwM2M(bootstrap)服务器的 URI,Payload 的安全模式,一些算法/密钥,服务器的短 ID 等信息
LwM2M 服务器(LwM2M Server)	1	服务器的短 ID,注册的生命周期,服务器对象的最小/最大周期,绑定模型等
访问控制(Access Control)	2	每个对象的访问控制权限
设备(Device)	3	设备的制造商、型号、序列号、电量、内存等信息
连接监控(Connectivity Monitoring)	4	网络制式、链路质量、IP 地址等信息
固件(Firmware)	5	固件包、包的 URI、状态、更新结果等
定位(Location)	6	经纬度、海拔高度、时间戳等
连接状态(Connectivity Statistics)	7	收集期间的收发数据量、包大小等信息

资源是一个逻辑概念,相当于一个对象实现一个功能所占用的资源。资源可以配置不同的权限:只读、或读写,可由操作 Access Control 对象进行维护。

接口的具体功能是由一系列的操作来实现的,LwM2M 的 4 种接口被分为上行操作和下行操作。

- 上行操作:LwM2M Client -> LwM2M Server。
- 下行操作:LwM2M Server-> LwM2M Client LwM2M Server。

使用设备管理和服务实现接口访问 LwM2M 客户端的对象实例和资源。该接口包括 7 种操作：创建、读数据、写数据、删除、执行、写属性和显示。

5.5.2 轻量级 M2M 协议栈

LwM2M 协议栈包括以下组成部分，如图 5-21 所示。

图 5-21 LwM2M 协议栈

(1) LwM2M 对象(LwM2M Objects)：每个对象都对应客户端的某个特定功能实体。LwM2M 规范定义了一下标准 Objects，比如，

urn:oma:lwm2m:oma:2；(LwM2M 服务器对象)；

urn:oma:lwm2m:oma:3；(LwM2M 访问控制对象)。

(2) LwM2M 协议(LwM2M Protocol)：定义了一些逻辑操作，比如读、写、执行、创建和删除。

(3) CoAP：是 IETF 定义的 Constrained Application Protocol 用来作为 LwM2M 的传输层，下层可以是 UDP 或 SMS。UDP 是必须支持的，SMS 是可选的。CoAP 有自己的消息头、重传机制等。

(4) DTLS：DTLS 协议为数据报协议提供通信隐私。该协议允许客户端/服务器应用程序以防止窃听、篡改或消息伪造的方式进行通信。DTLS 协议是基于传输层安全性(TLS)协议并提供等效的安全保证的协议。

5.5.3 LwM2M 体系架构

LwM2M 体系架构如图 5-22 所示。

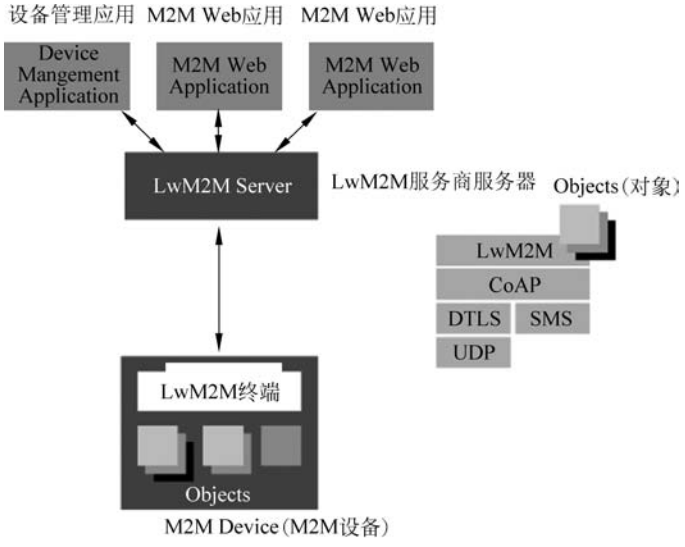

图 5-22 LwM2M 体系架构

一个终端至少可以接入一个服务商服务器，也可以接入多个服务商服务器。当接入多个服务器时需要进行选择操作，依据为引导(bootstrap)时获取的相关信息。

在引导完成后进行注册(registration)操作，如终端服务器交互工作模式：U(UDP)、

UQ(UDP with Queue Mode)、S(SMS)、SQ、US、UQS(UDP with Queue Mode and SMS)，而 UQSQ and USQ 在 V1.0.1 版本中不支持。

在注册完成后，才可启用设备管理与服务使能(Device Management and Service Enablement)及信息报告(Information Report)的接口服务，而这两类接口服务的主控方为服务器。如信息报告(Information Reporting)接口，网络可以通过观察(Observe)、停止观察(Cancel Observation)等操作来进行客户端 Notify 信息上报开关的控制。

5.6 小结

除了本章介绍的 MQTT、CoAP 和 LwM2M 协议之外，目前在物联网中常见的协议有十余种。表 5-10 对一些常见协议做了简要比较，限于篇幅，本书不对这些协议一一介绍，有兴趣的读者可以查阅资料进一步学习。

表 5-10 常见物联网协议对比

协议	DDS	MQTT	AMQP	XMPP	JMS	REST/HTTP	CoAP
抽象	Pub/Sub	Pub/Sub	Pub/Sub	NA	Pub/Sub	Requset/Reply	Requset/Reply
架构风格	全局数据空间	代理	P2P 或代理	NA	代理	P2P	P2P
QoS	22 种	3 种	3 种	NA	3 种	通过 TCP 保证	确认或非确认消息
互操作性	是	部分	是	NA	否	是	是
性能	1000 msg/s/sub	1000 msg/sub	1000 msg/s/sub	NA	1000 msg/s/sub	100req/s	100req/s
硬实时	是	否	否	否	否	否	否
传输层	默认为 UDP，TCP 也支持	TCP	TCP	TCP	不指定，一般为 TCP	TCP	UDP
订阅控制	消息过滤的主题订阅	层级匹配的主题订阅	队列和消息过滤	NA	消息过滤的主题订阅和队列订阅	N/A	支持多播地址
编码	二进制	二进制	二进制	XML 文本	二进制	普通文本	二进制
动态发现	是	否	否	NA	否	否	是
安全性	提供方支持，一般基于 SSL 和 TLS	简单用户名/密码认证，SSL 数据加密	SASL 认证，TLS 数据加密	TLS 数据加密	提供方支持，一般基于 SSL 和 TLS，JAAS API 支持	一般基于 SSL 和 TLS	

从当前物联网应用发展趋势看,MQTT 协议具有一定的优势。因为目前国内外主要的云计算服务商,比如阿里巴巴云、AWS、百度云、Azure 以及腾讯云均支持 MQTT 协议。另外,MQTT 协议比 CoAP 出现早,所以 MQTT 具有一定的先发优势。但随着物联网的智能化和多变化的发展,后续物联网应用平台肯定会兼容更多的物联网应用层协议。

习题

1. 列举你所知道的物联网的连接方式和通信协议,并简要叙述它们的基本情况。
2. NB-IoT 有哪些技术特点?
3. LoRa 与 LoRaWAN 有何联系?有何区别?
4. LoRaWAN 的网络架构是什么?
5. 简述 MQTT 协议的设计原则和优势。
6. 简述 MQTT 协议的数据包构成。
7. 简述 MQTT 协议的工作原理。
8. CoAP 协议的特点主要有哪些?
9. 试比较 MQTT 协议与 CoAP 协议的异同。
10. 简述 LwM2M 协议的构成。