

基于设备抽象层异构物联网平台适配框架研究

王玉东¹ 赵斌¹ 刘坤¹
WANG Yudong ZHAO Bin LIU Kun

摘要

随着物联网技术的快速发展,多样化的物联网平台在协议、数据格式及接口设计上的差异性,导致设备跨平台接入与管理的复杂性显著增加。为此,文章提出一种基于设备抽象层的跨平台适配框架,通过屏蔽底层设备的差异,实现在多种主流的物联网平台环境下的设备统一接入与跨平台互操作。实验结果表明,该框架在兼容性、复杂度等方面优于传统方案,有效降低平台迁移成本,为物联网设备跨平台协同提供了轻量化、高兼容性、低复杂度的解决方案。

关键词

物联网平台;设备抽象层;物联网设备;适配框架

doi: 10.3969/j.issn.1672-9528.2025.12.005

0 引言

随着物联网技术的快速发展,智能家居、工业物联网和智慧城市等典型应用场景中涌现出众多物联网平台(如 AWS IoT、Azure IoT、阿里云 IoT、私有云 IoT 等)^[1]。同时海量物联网设备(如传感器、智能终端、工业控制器等)接入物联网平台需求呈现爆发式增长^[2]。然而,当前物联网生态中存在多样化的平台架构、平台间的协议差异(如 MQTT、CoAP、HTTP、Modbus)、数据模型不兼容以及私有 API 壁垒,导致设备跨平台接入面临显著的兼容性问题^[3]。这种异构性使得开发者需要针对不同平台重复实现设备控制逻辑,极大增加了系统开发与维护成本^[4]。

当前研究中,设备抽象层(device abstraction layer, DAL)被广泛认为是解决异构性问题的有效途径^[5]。传统 DAL 方案多采用直接协议转换或单一总线的设计模式,在面对海量异构设备接入时显现出扩展性不足、实时性差等缺陷^[6]。因此,如何构建一种支持多平台无缝适配、具备弹性扩展能力的设备抽象层框架,成为亟待解决的问题^[7]。

本文给出一种基于 Apache Kafka 的分布式物联网设备抽象层适配框架。Kafka 作为高吞吐的分布式消息系统^[8],其发布订阅模型与持久化日志特性为设备与物联网平台异步通信提供了天然解决方案^[9]。

物联网设备通过 MQTT/CoAP 等原生协议接入设备抽象层,由设备抽象层统一转换为 Kafka 标准化消息^[10]。设备抽象层直接与物联网平台对接交互,为设备屏蔽平台差异,实现异构设备统一接入异构平台。

本研究在物联网场景中得到初步验证,为跨物联网平台的设备接入与管理提供了可复用的技术方案。

1 软件架构设计

物联网平台适配框架采用分层设计,自上而下分为物联网平台、设备抽象层、物联网设备层等,软件总体架构如图 1 所示。

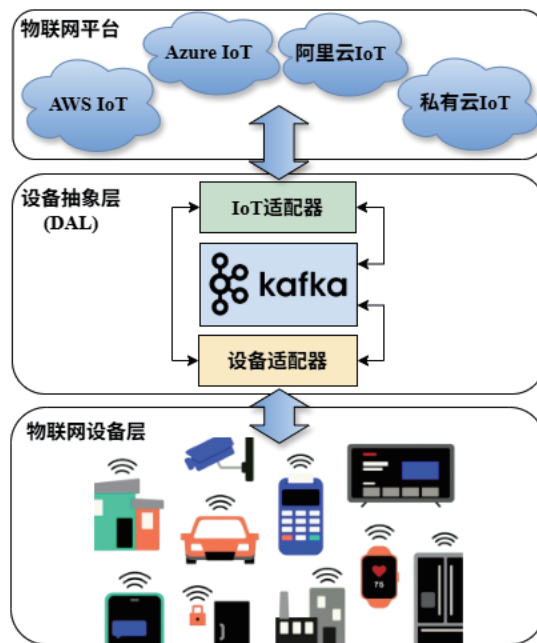


图 1 物联网平台适配框架

1.1 物联网平台

适配框架所适配的物联网平台主要包括主流的物联网平台:阿里云 IoT、AWS IoT、Azure IoT 以及一些私有云 IoT 等,这些主流的物联网平台的差异分析详情如表 1 所示。

1. 中国航空工业集团公司西安航空计算技术研究所
陕西西安 710000

表 1 主流物联网平台核心差异对比

平台	数据模型	通信协议	认证方式	设备管理
AWS IoT	Device Shadow (JSON)	MQTT、HTTP、WebSocket	X.509 证书、IAM Role	Thing Registry
阿里云 IoT	物模型 (TSL)	MQTT、CoAP、HTTP	三元组 (ProductKey Secret)	生命周期管理
Azure IoT	Digital Twin (DTDL)	MQTT、AMQP、HTTP	SAS Token、X.509 证书	Device Identity

平台的差异化直接影响设备抽象层的适配方案的实现细节以及功能的支持粒度。在设备抽象层屏蔽这些差异，为设备提供统一的接入与交互支持。

1.2 设备抽象层

设备抽象层是整个框架的核心，为物联网设备层提供统一的接入与管理接口，适配众多知名的物联网平台并与其交互。其模块自下而上主要有设备适配器、Kafka 分布式消息系统、IoT 适配器等，模块依赖关系如图 2 所示。

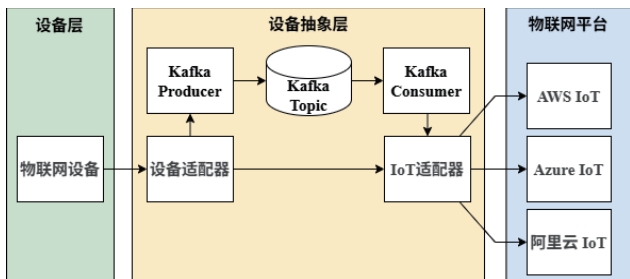


图 2 设备抽象层的模块依赖关系

1.2.1 设备适配器

设备适配器屏蔽底层设备的差异，与其他模块之间的依赖关系如图 3 所示，是适配框架中连接物联网设备与 Kafka、IoT 适配器等中间件的重要组件。设备适配器主要功能包括物联网设备接入适配和设备通信协议适配。

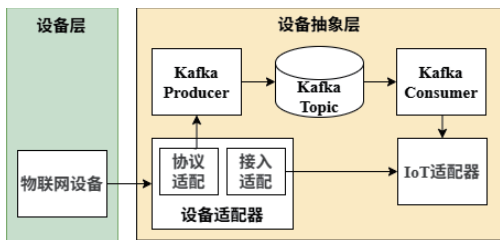


图 3 设备适配器与其他模块的依赖关系

接入适配模块为不同的设备提供统一的认证与注册接口。设备不感知平台的接入差异，在后续提到在设备层各类设备复用通用的认证与注册功能代码，统一对接设备适配器。IoT 适配器对接并适配物联网平台，对内提供了统一的接口，设备适配器会使用其接口最终完成设备与平台的间接认证与注册。

协议适配功能模块作为 Kafka 消息生产者代理，将来自不同物联网设备的协议数据转换为 Kafka 消息（JSON/Protobuf 格式），协议转换流程如图 4 所示。

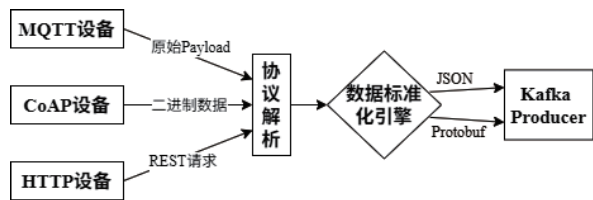


图 4 协议转换流程

转换后的协议消息由 Kafka 生产者发布至 Kafka 主题（Topics），协议适配详情如表 2 所示。

表 2 多协议适配与数据标准化

协议	适配逻辑	转换示例
MQTT	解析 Topic 层级结构； 处理 QoS 等级（0/1/2）； 支持 Retain 消息处理。	{ "t": 25.6 } → {"deviceId": "s1", "metrics": [{"name": "temp", "value": 25.6}]}
CoAP	解析 URI 路径和查询参数； 处理 CON/NON 消息类型； 支持 Observe 订阅模式。	二进制 Payload → Protobuf 编码
HTTP	适配 RESTful API (GET/POST/PUT/DELETE)； 处理 Header 中的认证信息。	GET /data?temp=25.6 → 标准化 JSON
Modbus	解析寄存器地址映射表； 处理大端 / 小端字节序。	寄存器 0x4001 的值 → "humidity": 45.2

1.2.2 Kafka 分布式消息系统

在设备抽象层中 Kafka 是设备适配器和 IoT 适配器之间的数据枢纽，解决异构设备与异构物联网平台间的数据流通过问题。设备适配器中的协议适配将不同协议（MQTT/CoAP）数据转换为 JSON/Protobuf 格式，写入 Kafka 主题中。Kafka 消费者将数据传递给 IoT 适配器，最终由 IoT 适配器转换为物联网平台特定协议。

Kafka 分布式消息系统作为核心中间件，承担着物联网设备数据缓冲、异步通信、流量削峰、跨平台数据分发等关键功能。为了适应物联网应用场景，需要对 Kafka 核心架构进行优化设计，优化详情如下：

（1）生产者（Producer）

数据写入 Kafka，支持批量发送，减少网络开销，通过数据压缩节省带宽。

（2）Broker 集群

存储和转发消息，集群部署提高可用性，SSD 优化存储。

（3）主题（Topic）

逻辑消息分类，按设备类型（如传感器/摄像头）、业务（如告警/遥测）划分 Topic。

(4) 分区 (Partition)

数据分片并行处理,按设备 ID 哈希分区,确保同一设备数据有序。

(5) 消费者组 (Consumer Group)

消费者组负载均衡,每个平台是独立的 Consumer Group,各自消费全量数据。

1.2.3 IoT 适配器

IoT 适配器是设备抽象层中的关键模块,主要负责从 Kafka 消费者获取标准化数据,通过 IoT 适配插件转换为目标平台 API 格式并推送至物联网平台,数据交互和模块依赖关系如图 5 所示。

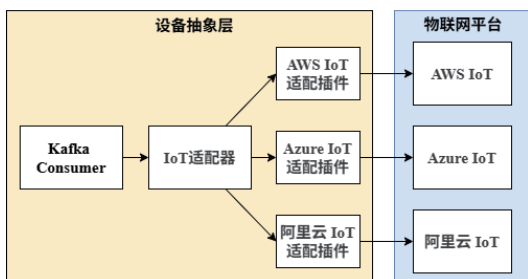


图 5 IoT 适配器的数据交互与模块依赖

IoT 适配器具体功能如下:

(1) 协议转换

将内部标准化数据(如 JSON)转换为目标平台专用协议(如阿里云物模型、AWS Shadow、Azure DTDL)。

(2) 屏蔽平台 API 差异

封装适配各平台 API(如认证、设备注册与管理、状态监控等 API),屏蔽平台 API 差异细节。平台差异对设备适配器以及 Kafka 是不可见的,对内提供统一的接口服务。

(3) 数据路由

根据业务规则将数据分发到指定平台(如医疗设备数据发往 A 平台,工业数据发往 B 平台)。

(4) 安全集成

集成并适配各平台认证、数据加密、访问权限控制等。

1.3 物联网设备层

物联网设备层是各类常见的物联网终端设备的集合。各类设备使用有线或无线方式联网,使用标准的物联网通信协议(如 MQTT、CoAP、HTTP)与设备抽象层对接交互。指令下行交互流程如图 6 所示。

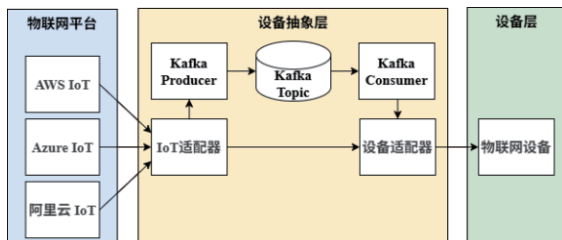


图 6 指令下行交互流程

开发者只需为此开发通用的接入与控制逻辑,各类设备复用这一通用的代码实现,从而不必为不同物联网平台和异构的物联网设备重复实现设备控制逻辑,实现一次开发,多设备复用。

2 实验验证

实验对比传统直接接入方案与本文给出的平台适配方案,初步验证物联网设备抽象层适配框架的兼容性、复杂度、迁移成本等核心指标。

2.1 实验环境

实验硬件环境如表 3 所示,软件环境详情如表 4 所示。

表 3 硬件环境

设备类型	数量	配置	通信协议	接入方式
智能温湿度传感器	5	ESP32 + DHT22	MQTT/CoAP	Wi-Fi/NB-IoT
智能灯泡	5	Raspberry Pi Pico + Relay	MQTT	Wi-Fi
工业网关	5	Jetson Nano + Modbus RTU	Modbus TCP/HTTP	有线以太网

表 4 软件与平台环境

组件	版本 / 配置	用途
阿里云 IoT Platform		
物联网平台	AWS IoT Core	多平台接入对比
	Azure IoT Hub	
消息中间件	Apache Kafka 3.2.0 (3 节点集群)	指令与数据中转
适配框架	本文给出的设备抽象层适配框架	设备适配 IoT 适配
对比方案	传统直接接入 (各平台原生 SDK)	作为对比基线

2.2 实验方案

实验从兼容性、复杂度、迁移成本 3 个维度设计实验方案,方案详情如表 5 所示。

表 5 实验方案设计

验证维度	具体场景	评估指标
兼容性	同一设备同时接入阿里云、AWS、Azure 等物联网平台。	平台功能支持率、设备功能支持率。
复杂度	新增设备类型的开发工作量。	代码修改行数、配置项数量。
迁移成本	从阿里云迁移至 AWS 的改造时间。	人工小时数、API 调用变更比例。

2.3 实验结果

按照此前实验方案进行测试验证,兼容性实验结果如表

6 所示，复杂度对比结果详如表 7 所示。

表 6 兼容性实验结果

平台组合	平台功能支持率 /%	设备功能支持率 /%
阿里云 +AWS	98.7	100
AWS + Azure	97.2	98
三平台同时接入	96.5	97

表 7 复杂度对比结果

指标	传统方案	本文方案	降低幅度
新增设备代码量	200~300 行	50~80 行	↓ 约为 70%
平台迁移配置项	15~20 项	3~5 项	↓ 约为 75%

2.4 实验结论

支持多平台并行接入，平台功能支持率大于 96%，显著优于传统单平台绑定方案。同一设备既可以向阿里云上报数据，同时也可以通过 AWS IoT 控制器实现跨平台协同。

适配新型平台时，只需开发 1 个 IoT 适配插件。新增设备类型只需复用已有的接入和交互代码实现与设备抽象层对接，无需为每个平台重复编码，代码量减少约 70%。平台迁移只需修改配置文件中 IoT 适配器类型和平台认证方式等，配置项减少约 75%。

3 结语

本文针对物联网设备跨平台协同的兼容性差、迁移成本高、开发复杂等问题，给出了一种基于物联网设备抽象层适配框架，为设备屏蔽了物联网平台的差异，实现了平台与设备的解耦设计。物联网设备只与设备抽象层对接交互，不感知平台的差异。适配新型物联网平台，只需在设备抽象层实现 IoT 适配插件。管理新型物联网设备只需复用已有的接入与控制逻辑代码对接设备抽象层，不必为资源受限的物联网设备开发维护多套平台对接交互代码。

实验结果表明，该适配框架能大幅度降低了集成开发与维护成本，减少了约 70% 的开发工作量，有效地降低了异构设备对接异构平台的复杂度。本研究的分层设计和插件化平台适配架构不仅适用于现有主流平台，还可快速扩展至新兴物联网平台，为物联网异构设备跨平台协同提供了一种轻量化、高兼容性、低复杂度的解决方案。

参考文献:

[1]YACCHIREMA D, PALAU C. Interworking of Onem2M-based IoT systems and heterogeneous IoT devices[C]//IEEE 2020 XLVI Latin American Computing Conference (CLEI). Piscataway:IEEE,2020: 262-267.

[2]NOGUEIRA CAVALCANTI RIBEIRO A M, DO CARMO P R X , SADOK D,et al. Comparative analysis of abstraction layer performance of IoT devices management platforms[C] // IEEE 2019 6th International Conference on Internet of Things: Systems, Management and Security (IOTSMS). Piscataway: IEEE, 2019: 35-42.

[3]KULATUNGA N A, SAMPAIO D J B S. IoT platforms and hardware integrations for industry 4.0 applications[C] //IEEE 2019 14th Conference on Industrial and Information Systems (ICIIS). Piscataway:IEEE,2019: 209-214.

[4]RAZZAQUE M A, MILOJEVIC-JEVRIC M, PALADE A,et al. Middleware for internet of things: a survey[J].IEEE internet of things journal. 2016,3(1): 70-95.

[5]HASSAN H H, BOULOUKAKIS G, SCALZOTTO L, et al. A message broker architecture for adaptive data exchange in the IoT[C]//IEEE 21st International Conference on Software Architecture Companion (ICSA-C). Piscataway:IEEE,2024: 151-158.

[6] Use Apache Kafka on HDInsight with Azure IoT Hub [EB/OL].(2024-11-13)[2025-05-17].<https://learn.microsoft.com/en-us/azure/hdinsight/kafka/apache-kafka-connector-iot-hub>.

[7]Communicate with an IoT hub using the MQTT protocol [EB/OL].(2025-03-21)[2025-06-22].<https://learn.microsoft.com/en-us/azure/iot/iot-mqtt-connect-to-iot-hub>.

[8]AWS IoT core developer guide[EB/OL]. (2025-03-23) [2025-06-20].<https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html>.

[9] 阿里云 . 云端开发指南 [EB/OL]. (2024-06-03)[2025-06-23]. <https://help.aliyun.com/zh/iot/developer-reference/developer-guide>.

[10]Kafka 4.0 documentation[EB/OL]. (2025-03-18)[2025-05-20].<https://kafka.apache.org/documentation>.

【作者简介】

王玉东（1985—），男，辽宁宽甸人，硕士，软件研发工程师，研究方向：嵌入式软件。

赵斌（1993—），男，陕西眉县人，硕士，研究方向：嵌入式软件。

刘坤（1997—），女，山东临邑人，硕士，工程师，研究方向：嵌入式软件。

（收稿日期：2025-06-25 修回日期：2025-12-04）