ICS 35.240.99
CCS L70

团 体 标 准

T/CABEE 130-2026

智慧建筑控制系统数据通信技术要求

Technical requirements for data communication in intelligent building control systems

2026-03-05 发布 2026-05-01 实施

中国建筑节能协会 发布

目 次

前　言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。本文件的协议实现可参照 GB/T 28847.6 进行一致性测试。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由中国建筑节能协会提出并归口。

本文件起草单位：复旦大学、中国建筑国际集团有限公司、青岛海尔智慧楼宇科技有限公司、北京和欣运达科技有限公司、江苏英特信智能科技有限公司、上海美控智慧建筑有限公司、中邮建技术有限公司、无锡研奇智联技术有限公司、广州邮电通信设备有限公司、中外建设信息有限责任公司、北京易住行技术有限公司、浙江大学、山东省医药工业设计院有限公司、青岛海信智慧建筑科技有限公司、深圳海智创科技有限公司、深圳数智乐居科技有限公司、深圳广物智酷咨询有限公司、深圳威尔扬科技有限公司、知真行远数字技术（杭州）有限责任公司。

本文件主要起草人：周小林、郑立荣、付晶、张海鹏、周威斌、张青、唐俊杰、马虹、刘政、张海洋、李雁飞、阎杰、周斌、沈阳、王莎、尚治宇、李野、谢立、付小晓、张琨、王冲、左壮、廖景明、李露露、尹志强、包侃侃。

本文件主要审查人：张桂青、郝斌、杨亚龙、聂秀英、周小平、庄园、王宝龙。

智慧建筑控制系统数据通信技术要求

1 范围

本文件规定了智慧建筑控制系统中数据通信的技术要求（以下简称BACnetMQ），包括智慧建筑数据通信标准基本网络模型与协议架构、数据传输服务、网络传输要求、网络安全机制等内容。

本文件适用于智慧建筑控制系统中的设备间数据通信设计及实现。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 28847.5 建筑自动化和控制系统 第5部分：数据通信协议

GB/T 28847.6 建筑自动化和控制系统 第6部分：数据通信协议一致性测试

GB/T 32907-2016《信息安全技术 SM4分组密码算法》

GB/T 32918.1-2016《信息安全技术 SM2椭圆曲线公钥密码算法 第1部分：总则》

GB/T 32918.2-2016《信息安全技术 SM2椭圆曲线公钥密码算法 第2部分：数字签名算法》

GB/T 32918.3-2016《信息安全技术 SM2椭圆曲线公钥密码算法 第3部分：密钥交换协议》

GB/T 32918.4-2016《信息安全技术 SM2椭圆曲线公钥密码算法 第4部分：公钥加密算法》

GB/T 32918.5-2016《信息安全技术 SM2椭圆曲线公钥密码算法 第5部分：参数定义》

3 术语和定义

下列术语和定义适用于本文件。

3.1

楼宇自动化和控制网络 BACnet building automation control network protocol

用于楼宇自动化和控制系统设备间数据通信的开放式协议

3.2

消息队列 MQ message queue

一种用于在分布式系统组件之间异步传递消息的通信中间件模型。

3.3

智慧建筑控制系统数据通信消息队列协议 BACnetMQ building automation control network protocol message queue protocal

基于BACnet数据结构与消息队列技术融合形成的、适用于智慧建筑控制系统的数据通信协议。

3.4

Mesh网络 Mesh network

一种网络拓扑结构，其中每个节点都可以与一个或多个对等节点直接通信，并具备自组织和自修复能力。

3.5

访问控制列表 ACL access control list

用于定义系统或网络设备对用户、进程或数据访问权限的规则集合。

3.6

通用唯一标识符 UUID universally unique identifier

一种标准化的标识符格式，用于在分布式系统中全局唯一地标识某个实体或信息。

3.7

路由表 routing table

存储网络中各目标地址与对应转发路径（如下一跳地址）映射关系的数据结构。

4 缩略语

下列缩略语适用于本文件。

ACL：访问控制列表（Access Control List）

AMQP：高级消息队列协议（Advanced Message Queuing Protocol）

APDU：应用层协议数据单元（Application Protocol Data Unit）

BAC：建筑自动化与控制（Building Automation and Control）

BACnet：楼宇自动化与控制网络（Building Automation and Control Network）

BVLC：BACnet虚拟链路控制（BACnet Virtual Link Control）

dst：消息目标地址（Destination Address）

IP：互联网协议（Internet Protocol）

MAC：媒体访问控制地址（Media Access Control Address）

Mesh：网状网络（Mesh Network）

MQTT：消息队列遥测传输协议（Message Queuing Telemetry Transport）

MSTP：主从/令牌传递（Master-Slave/Token-Passing）

NPDU：网络协议数据单元（Network Protocol Data Unit）

SM2：非对称加密算法（Chinese SM2 Cryptographic Hash Algorithm）

SM3：密码杂凑算法（Chinese SM3 Cryptographic Hash Algorithm）

SM4：分组密码算法（Chinese SM4 Block Cipher Algorithm）

src：消息源地址（Source Address）

RREP：路由回复（Route Reply）

RREQ：路由请求（Route Request）

TLS：传输层安全协议（Transport Layer Security）

UUID：通用唯一标识符（Universally Unique Identifier）

5 智慧建筑控制系统组成

5.1 系统架构

智慧建筑控制系统宜采用分层架构设计，系统组成如图1所示，采用四层逻辑架构：管理应用层、通信网络层、边缘接入层、设备感知层。

各功能层间通过符合本文件规定的BACnetMQ协议实现标准化数据通信，该协议支持UUID寻址、Mesh组网、消息队列传输等特性，并保持与传统BACnet系统的兼容。

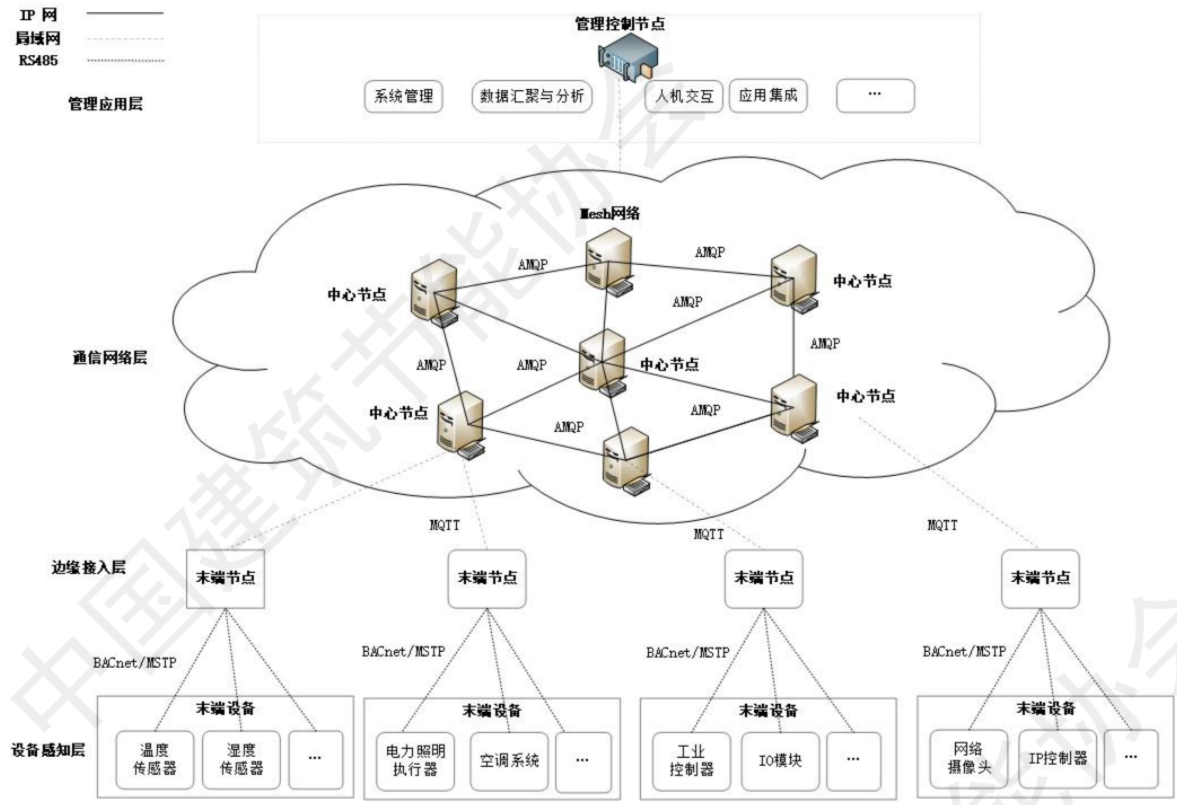

图 1 智慧建筑控制系统架构

5.2 系统各部分功能

5.2.1 管理应用层

部署建筑设备监控、能源管理、运维管理等应用系统，通过标准接口与通信网络层交互，应具备以下功能：

a) 系统管理：负责系统的配置、监控、策略下发与全局状态维护。

b) 数据汇聚与分析：接收来自通信网络层的数据，进行分析、存储与可视化展示。

c) 人机交互：为操作人员提供图形化界面，实现对整个系统的监视与控制。

d) 应用集成：提供标准接口，支持与其他信息系统的数据交互。

5.2.2 通信网络层

部署实现BACnetMQ协议的消息中间件或通信平台，由多个中心节点通过AMQP协议构成Mesh网络，是系统数据通信的核心，应具备以下功能：

a) 消息路由与转发：基于UUID实现跨节点、跨网络的消息寻址与投递。

b) 协议转换与适配：支持MQTT协议和AMQP协议的接入与处理。

c) 网络管理与维护：实现Mesh网络的自组织、自修复与路由优化。

d) 安全通信保障：提供传输加密、身份认证、访问控制等安全机制。

本文件所规定的BACnetMQ协议及相关技术要求主要适用于本层。

5.2.3 边缘接入层

由部署于建筑现场的末端节点组成，直接连接传感器与执行器，具备以下功能：

a) 数据采集与执行：采集环境参数（如温度、湿度）或执行控制指令（如开关阀门）。

b) 本地逻辑处理：可执行预设的本地控制逻辑与告警联动。

c) 协议接入与转换：通过BACnet MSTP、BACnet/IP等现场协议连接末端设备，并将其转换为MQTT协议格式，通过MQTT协议接入通信网络层。

d) 断网续传：在网络中断时缓存数据，恢复后自动续传。

5.2.4 设备感知层

包括各类末端设备，如传感器、执行器、控制器、仪表等，如温度传感器、湿度传感器、照明执行器、空调控制器、IO模块、网络摄像头、IP控制器等，通过标准工业总线（如BACnet MSTP）或网络协议（如BACnet/IP）接入系统。

5.3 控制系统数据通信场景

智慧建筑控制系统中的数据通信主要包括但不限于以下场景：

a) 监测数据上行：设备感知层传感器采集数据，通过现场协议上传至所属边缘接入层末端节点；末端节点将数据封装为MQTT消息，发布至通信网络层；通信网络层通过Mesh网络将数据路由并转发至管理应用层进行处理与展示。该场景要求通信协议支持轻量级、低功耗的发布/订阅机制。

b) 控制指令下行：管理应用层生成控制指令，通过通信网络层封装为MQTT消息，根据目标UUID经Mesh网络路由至相应中心节点；中心节点将指令下发至目标边缘接入层末端节点；末端节点转换协议格式后驱动设备感知层执行器完成动作。该场景要求通信协议支持可靠、低时延的点对点或组播通信，并具备优先级与确认机制。

c) 跨子系统联动：某一子系统的报警事件通过边缘接入层上报至通信网络层；通信网络层根据联动规则，将事件消息路由至相关子系统关联的中心节点，触发联动控制指令下发。该场景要求通信协议支持基于内容的灵活路由与跨网段事件转发。

d) 设备发现与网络自组织：新设备接入时，通过边缘接入层向通信网络层注册；通信网络层利用Mesh网络自组织特性，动态更新路由表。网络故障时，Mesh网络可自动检测并修复。该场景要求通信网络层支持动态寻址、发现与自修复能力（见第7章）。

6 智慧建筑数据通信标准基本网络模型与协议架构

6.1 BACnetMQ 简化架构模型

智慧建筑数据通信协议（BACnetMQ）基于 GB/T 28847.5 所规定的楼宇自动化和控制网络（BACnet）数据模型与通信框架。该协议通过在传输层之上构建虚拟化通信层次并集成消息队列机制，旨在克服传统BACnet/IP在跨子网通信、可靠性及数据单元长度等方面的限制。

BACnetMQ的简化协议架构模型如图2所示。其核心是在OSI模型的传输层协议之上，重构出四层虚拟化通信层次，并通过将BACnet协议数据封装于MQTT或AMQP消息之中，实现BACnet系统与现代消息队列网络的无缝集成。

图2 BACnetMQ协议架构模型

6.1.1 虚拟数据链路层

虚拟数据链路层应实现基于消息队列的可靠数据传输机制，并满足以下要求：

a) 消息保障：应提供消息缓冲机制，确保在网络拥塞或设备暂时失效时消息不丢失。

b) 顺序与延迟控制：应保证消息按发送顺序可靠传递，并优化传输路径以最小化通信延迟。

c) 异步通信：应支持异步通信模式，允许设备在无需等待即时确认的情况下进行消息的发送与接收。

6.1.2 虚拟网络层

虚拟网络层应基于高级消息队列协议实现网络级的通信操作，并满足以下要求：

a) 协议原语：应通过实现AMQP协议的核心指令集，为网络操作提供标准化的通信原语。

b) 消息交换模式：应支持点对点、发布-订阅及请求-应答等多种消息交换模式。

c) 网络适配：应能适配局域网、广域网等不同网络环境，确保通信的灵活性与可扩展性。

6.1.3 虚拟传输层

虚拟传输层应负责基于实时网络状态的数据路由，并满足以下要求：

a) 动态路由：应建立并维护动态路由表，根据网络拓扑与状况实时更新路由信息。

b) 路径优化：应基于路由表为数据包选择最优传输路径，确保高吞吐量与低延迟。

c) 可靠投递：应保障数据包在网络中可靠、准确地路由至目标节点。

6.1.4 虚拟应用层

虚拟应用层应负责BACnet数据的封装、映射与交互，并满足以下要求：

a) 数据封装：应将BACnet应用协议数据单元、对象及服务正确封装到BACnetMQ协议的消息格式中。

b) 语义保持：在封装与转换过程中，应保持BACnet对象模型与服务语义的完整性与一致性。

c） 协议转换：应实现BACnet数据与消息队列协议格式间的双向转换，以支持采用不同通信协议的设备或系统间的互操作。

6.2 BACnetMQ 协议的角色与定位

BACnetMQ是本文件规定的智慧建筑控制系统数据通信协议，作为各功能层间的标准化通信接口，主要实现以下通信：

a）通信网络层内部节点间的数据交换：中心节点间采用AMQP协议构成Mesh网络，实现高可靠、多路径的数据交换。

b）通信网络层与边缘接入层间的上下行通信：通过MQTT协议的发布/订阅机制实现，边缘接入层作为MQTT客户端，通信网络层提供MQTT代理服务。

c）通信网络层与管理应用层间的数据接口：通过RESTful API、WebSocket或专用协议接口实现。

d）与传统BACnet系统的双向兼容通信：通过在边缘接入层实现BACnet协议到BACnetMQ协议的转换，保持与传统BACnet设备（BACnet MSTP和BACnet/IP）的兼容性。

6.3 BACnetMQ 协议关键特性与设计要求

6.3.1 MQTT/AMQP 节点唯一标识

每个节点具备唯一的UUID，用于区分和标识节点，以确保通信过程的错误与重复避免。

6.3.2 Mesh 网络结构和自组网自修复能力

基于MQTT/AMQP节点实现自组网和自修复的Mesh网络架构，提高网络的可靠性和应急能力。

6.3.3 Mesh 节点路由表维护

在AMQP节点中建立和维护Mesh路由表，实施节点之间的最优路由策略，确保数据包的高效传输。

6.3.4 BACnet 数据封装和发布

将BACnet数据结构封装为符合MQTT/AMQP协议格式的消息，并通过定义的消息并发上传到指定的交换机/消息队列。

6.3.5 消息转发和路由管理

Mesh网络根据路由表中定义的转发规则，将消息并发到目标节点的接收消息队列，确保数据包在网络中的高效和最优通迁。

6.3.6 数据解析和还原

节点分析接收的消息，并将BACnet数据结构还原为原始格式，便于后续处理。

6.3.7 BACnet 设备接收数据和操作执行

节点将还原后的BACnet数据传递给相关BACnet设备，实现接收数据后的监测和操作执行功能。

6.3.8 BACnetMQ 设计要求

该架构应允许使用MQTT/AMQP作为传输协议，通过构建虚拟化Mesh网络，将BACnet数据结构转换为应用层消息，并在订阅端进行解析和还原，完成BACnet和MQTT/AMQP之间的数据交互和传输。进一步实现时，应根据硬件资源和网络资源选择适合的MQTT/AMQP协议作为通信基础。

7 智慧建筑网络传输要求

7.1 BACnetMQ 网络拓扑

BACnetMQ的实现架构应基于AMQP节点构建Mesh网状逻辑结构，以提升系统的扩展性、可靠性和通信效率。该拓扑结构应满足智慧建筑控制系统的复杂数据通信需求，支持端到端的多信道通信、动态路径选择和高效数据传输，并具备高扩展性与容错能力。

BACnetMQ网络采用分层拓扑设计，如图3所示，主要包括：

a）骨干网层（IP骨干网）：由多个中心节点（AMQP Node） 通过AMQP协议互联，构成一个跨越IP网络的Mesh网络，实现高可靠、多路径的数据路由与转发。

b）接入层（如LAN1、LAN2）：各本地网络中的末端节点（MQTT Node） 通过MQTT协议接入最近的中心节点，实现轻量级设备的数据上传与控制指令接收。

c）协议兼容层：系统通过协议转换与封装技术，兼容传统的BACnet/IP和BACnet/MSTP设备及通信模式。。

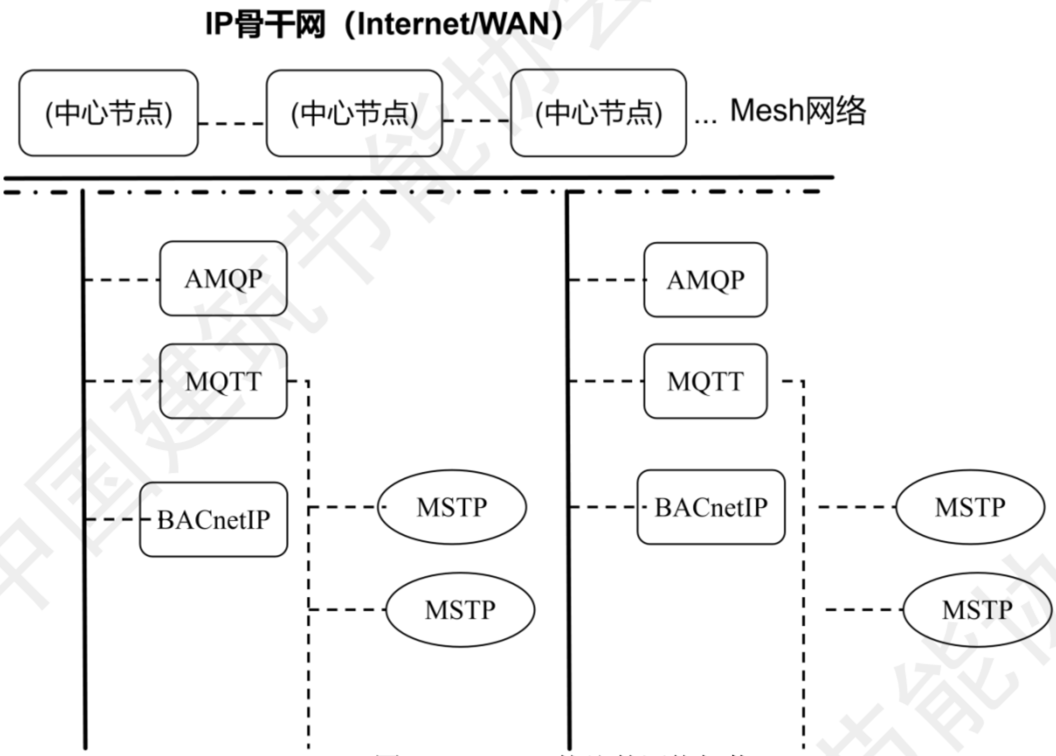

图3 BACnetMQ 协议的网络拓扑

7.1.1 协议支持要求

系统应支持同时使用AMQP协议与MQTT协议，以满足不同设备和网络环境的通信需求。

7.1.2 MQTT 协议应用

末端低算力设备或传感器应通过MQTT协议接入系统，并通过MQTT协议将消息发布至中心节点的AMQP消息路由网络，以实现消息的高效转发与处理。

7.1.3 AMQP 协议应用

对等的中心节点之间应使用AMQP协议建立通信信道，并在基于虚拟化的消息路由网络中实现数据交互、消息转发及局部广播功能。

7.1.4 传统协议兼容性

系统应兼容BACnet/IP及BACnet/MSTP通信模式，并通过协议转换与封装技术实现与传统BACnet设备的无缝通信。

7.1.5 中心节点

中心节点应作为系统的通信中枢，并应具备以下能力：

a）应与其他中心节点连接，构成一个虚拟化的通信网络；

b）应为所在区域内的普通MQTT节点提供该虚拟通信网络的接入服务。

7.1.6 末端节点

末端节点承担设备数据的接入与双向通信功能，应支持以下特性：

a）末端节点通过MQTT协议接入网络，发布设备数据并订阅控制指令，实现轻量级通信需求。

b）末端节点通过协议转换与数据封装技术，实现BACnet设备与MQTT协议之间的互联互通，可无缝 适配传统BACnet/IP和BACnet/MSTP通信模式。

c）末端节点将BACnet消息封装为符合MQTT协议格式的消息，发布至指定的服务端（公共消息队列）。同时，节点基于自身UUID订阅唯一标识的TOPIC频道，用于接收控制指令或数据更新。该机制确保BACnet数据包的可靠双向通信。

7.1.7 节点关键参数

末端节点应配置以下关键参数以保证通信的稳定性和一致性。

a）每个节点应配置唯一标识符（UUID），支持英文字母、数字和下划线（_）。UU ID长度应不超过36个字符，用于标识节点的身份，确保在通信网络中的唯一性。

b）节点应设置心跳时间以保持与MQTT代理的连接活跃。推荐保活时间为300s以上，在网络条件 不稳定的环境下，可适当延长心跳时间以提高连接可靠性。

c）节点应指定用于消息发布和订阅的MQTT主题（Topic）。支持使用通配符（如 # 和 +）实现批 量操作功能，以满足复杂主题过滤需求。

7.1.8 协议的节点逻辑拓扑架构

图4描述了基于**BACnet MQ**协议的节点逻辑拓扑架构，展示了MQTT、AMQP、BACnet/IP、MSTP协议之间的交互关系及其适用场景。图中采用虚线和实线区分了不同协议之间的连接方式。实线表示末端节点（MQTT）与中心节点（AMQP）的连接。虚线表示中心节点间的AMQP信道，支持数据转发与同步。

BACnetMQ节点架构结合了传统的BACnet协议（如BACnet/IP和BACnet/MSTP）与消息队列协议（MQTT和AMQP），实现了楼宇控制系统中的多协议互联互通。该架构利用AMQP的高可靠性与MQTT的轻量级特点，支持复杂的网络拓扑和高效的数据交互。

AMQP协议用于实现中心节点之间的高可靠性通信。节点间通过AMQP建立信道，可支持消息的路由、转发和局部广播。

MQTT协议用于连接低算力设备（如传感器）和末端节点。末端设备通过MQTT将消息上传至中心节点的AMQP网络，并从中心节点订阅控制指令。

通过协议转换技术，实现BACnet/MSTP设备的接入，保证传统设备与现代消息网络的无缝连接。该拓扑适用于楼宇自动化系统中的设备集成和多网络场景。

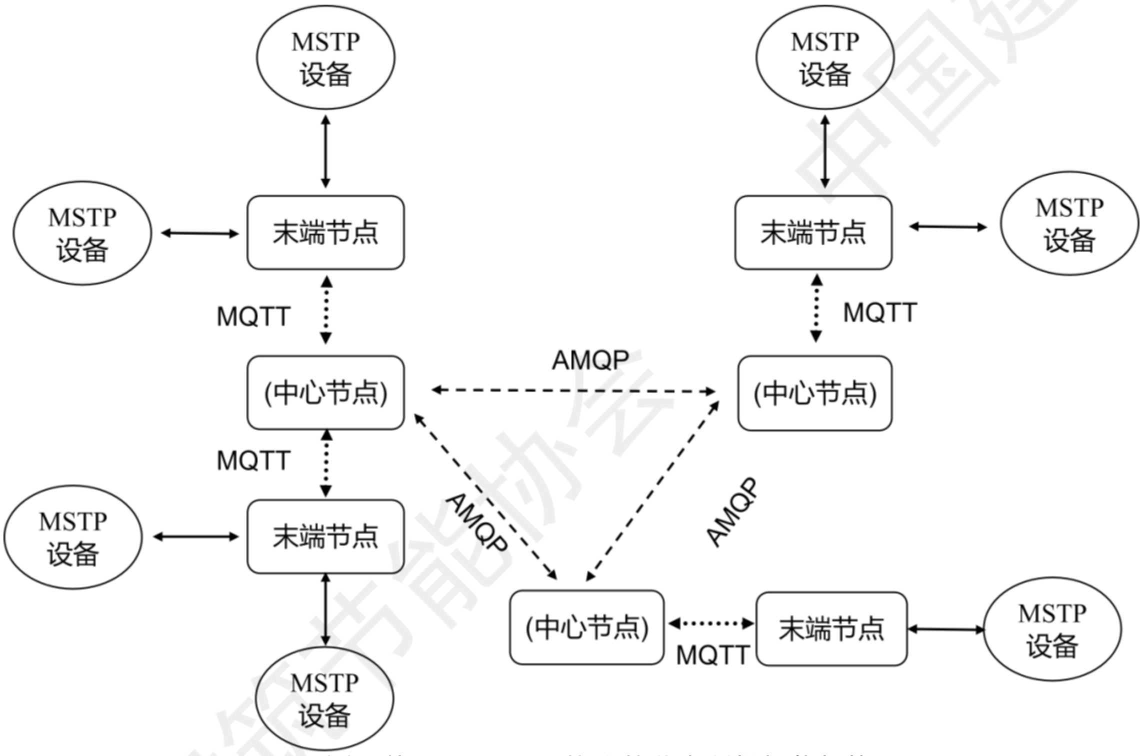

图4 基于BACnetMQ 协议的节点逻辑拓扑架构

7.1.9 唯一标识 UUID

基于BACnetMQ的架构突破了传统BACnet/IP的容量限制，通过虚拟化技术实现了设备在大规模网络中的高效通信。该架构消除了网段和IP地址对BACnet 协议的限制，无需依赖BBMD（BACnet/IP 广播管理设备）机制即可实现跨网段的点对点通信，支持海量设备在同一网络下的共存和高效管理。

为确保在MQ虚拟化网络中对设备的唯一标识性，规范要求基于设备的MAC地址生成唯一标识符（UUID）。UUID的生成能够满足全球范围内的唯一性要求，同时具备灵活性和可扩展性。

7.1.9.1 规范要求

在BACnetMQ的架构下，为确保设备在网络中的唯一性标识，约定使用 UUID（基于 MAC 地址）。具体生成方式采用 UUID1 算法，通过结合设备的 MAC 地址、时间戳和随机数生成唯一标识符，满足以下需求：

a)唯一性：在全球范围内唯一，适合大规模分布式系统。

b)可追溯性：通过结合设备 MAC 地址，实现标识与物理设备的绑定。

c)兼容性：与现有网络协议和设备命名体系兼容，便于扩展。

7.1.9.2 UUID版本及生成算法

UUID（通用唯一标识符）提供了五种主要版本的生成算法，根据应用场景的不同特点选择适用的版本：

a) UUID1（基于时间戳）

通过当前时间戳、设备 MAC 地址和随机数生成。全球范围内唯一，适用于对时间敏感的场景。

b) UUID2（基于分布式计算环境 DCE）

算法类似于 UUID1，将时间戳的前 4 位替换为 POSIX UID（用户标识符）。适用于分布式计算环境的特定应用场景。

c) UUID3（基于名字的 MD5 散列值）

通过对命名空间和名字进行 MD5 散列计算生成。保证同一命名空间内名字的唯一性，以及跨命名空间的唯一性。

d）UUID4（基于随机数）

使用伪随机数生成 UUID，无需依赖设备信息或时间戳。重复概率极低，可满足大多数通用应用场景。

e）UUID5（基于名字的 SHA-1 散列值）

与 UUID3 类似，但使用更安全的 SHA-1 散列算法。在安全性和唯一性要求更高的场景中适用。

7.2 Mesh 网络

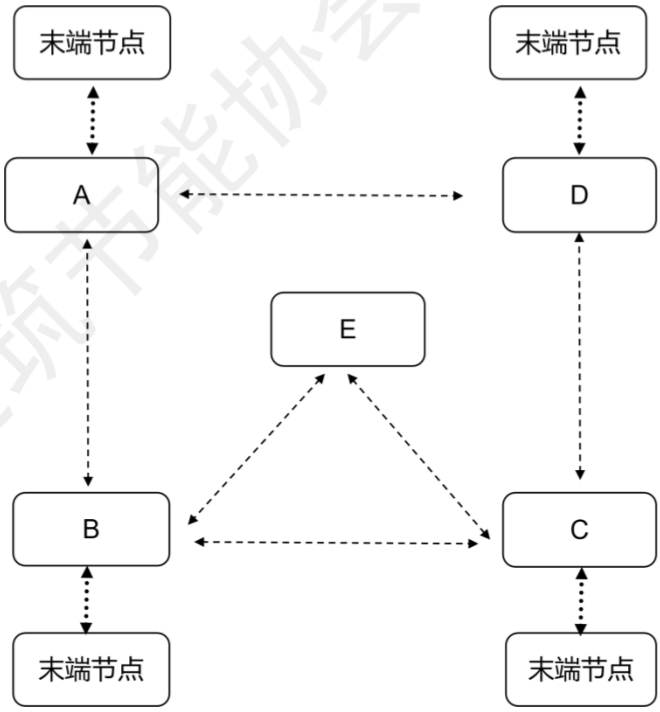

图5 中心节点Mesh 网络拓扑结构

7.2.1 Mesh 网络定义与架构

Mesh网络应是一种基于中心节点互联、具备自组网能力的动态网状拓扑结构。其网络拓扑结构示意图如图5所示。

7.2.2 多路径通信

Mesh网络应支持节点间的多路径通信。例如，从中心节点A到中心节点C可存在诸如A→B→C、A→D→C、A→B→E→C等不同传输路径。多路径设计旨在提升网络的可用性与稳定性。

7.2.3 规范要求

Mesh网络的设计应符合下列要求：

a）**拓扑设计：**Mesh 网络应采用动态路由机制，支持多跳互连和自适应路径选择，以适应复杂物理环境和拓扑动态变化的需求。

b）**设备特性：**加入 Mesh 网络的设备应具备自动发现和自组织能力，并能够兼容网络的动态路由和自动修复机制。

c）**可靠性目标：**网络设计应满足高可靠性要求，确保在链路故障或节点失效情况下，网络仍能维持通信功能，恢复时间不超过预定限值。

7.3 消息队列和订阅

7.3.1 中心节点与末端节点通信

7.3.1.1 规范要求

a) 消息队列部署：所有消息队列应统一部署在中心节点，并以 UUID为唯一标识，以支持分布式设备的高效通信。

b) 末端节点访问：末端节点仅应通过 HTTP、AMQP 等协议访问中心节点的消息队列，实现对消息的发送与订阅。

c) 容错与可靠性：中心节点应具备消息缓存与重传能力，确保在网络延迟或故障时，消息不会丢失。

7.3.1.2 适用场景

该消息队列与订阅机制适用于智慧建筑中高密度设备接入和复杂控制需求的通信场景，其轻量级特性能够适配资源受限的末端节点。

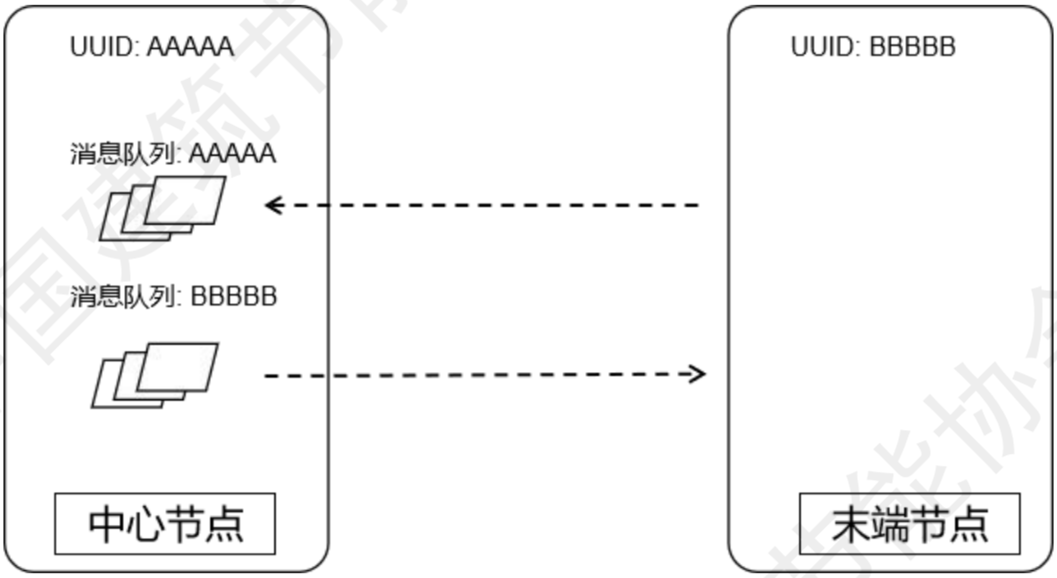

图 6 中心节点与末端节点的消息队列交互

在智慧建筑控制系统中，中心节点与末端节点的通信采用基于消息队列的双向通信机制，以满足末端节点资源有限的设备或网络环境的需求。具体通信流程和机制如下：

a) 所有消息队列部署于中心节点，末端节点通过访问中心节点的消息队列完成消息的发送与订阅。

中心节点为每个设备分配以其 UUID命名的消息队列，用于统一接收发送给该设备的消息。中心节点与末端节点间的消息队列交互模型，如图6所示。

b) 末端节点通过访问中心节点中以自身 UUID为 ID 的消息队列，完成消息的订阅与消费，适配低功耗、低计算能力的设备场景。

c) 中心节点维护以末端节点 UUID为 ID 的消息队列，用于辅助消息的缓存与订阅服务。

末端节点仅应订阅与自身相关的消息队列，无需额外资源开销，从而简化通信逻辑并提升效率。

7.3.2　中心节点与中心节点通信

7.3.3.1 规范要求

a）唯一标识：所有中心节点的消息队列应以节点的 UUID命名，确保队列的唯一性与可追溯性。

b）队列交互机制：中心节点应支持将消息发布到目标节点的队列中，同时订阅和消费自身队列中的消息，以实现高效的双向通信。

C）可靠性保障：消息队列应具备持久化和重传机制，以应对通信中的网络波动或节点故障。

7.3.3.2 应用场景

该中心节点与中心节点的通信机制适用于智慧建筑中应跨区域、跨子系统的控制与协调场景，例如多楼层间的设备协调控制、分布式系统间的状态同步与指令下发、高可靠性场景中的冗余中心节点通信等。

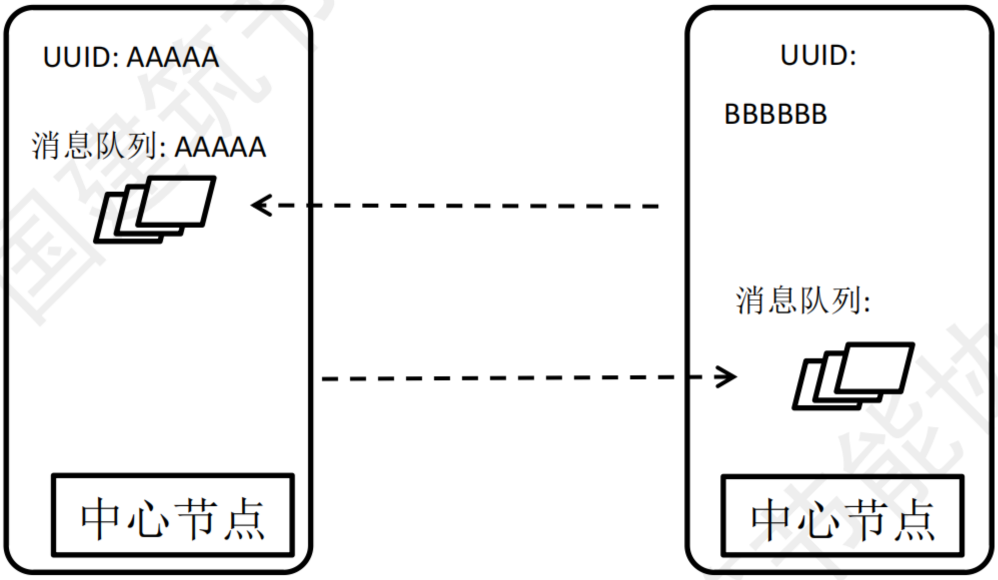

图 7 中心节点与中心节点的消息队列交互

在智慧建筑控制系统中，当多个中心节点之间应进行双向通信时，采用基于消息队列的机制实现高效、可靠的交互。

a）消息队列声明

每个中心节点声明一个以自身 UUID为唯一标识的消息队列，用于接收发往本节点的消息。

各中心节点通过订阅自己的消息队列，实现对消息的接收与消费。中心节点与中心节点之间的消息队列交互设计，如图7所示。

b）消息发布与双向通信

在双向通信过程中，中心节点将消息发布到对方的 UUID消息队列中。

每个中心节点在消费自己的队列时，即可获取来自其他节点的消息，从而完成双向通信。

c）通信流程

节点 A 向节点 B 通信时，节点 A 将消息发布到以节点 B UUID命名的消息队列中。节点 B 订阅并消费该队列中的消息。同理，节点 B 向节点 A 通信时，将消息发布到节点 A 的消息队列中，节点 A 消费队列完成接收。

通过这种设计，任意两个中心节点之间都可以实现点对点通信，无需额外的中间代理或复杂的路由逻辑。

7.3.3 备用交换机（Alternate Exchange）

备用交换机是用于处理无法路由到当前交换机任何队列的消息的机制。当消息在当前交换机中无法找到匹配的路由时，这些消息会被重新路由到备用交换机。通过配置备用交换机的路由逻辑，可以灵活地对未路由消息进行管理，例如将消息重定向到一个特殊的 死信队列（Dead-Letter Queue）。

7.3.4 死信队列（Dead-Letter Queue）

死信队列是一种特殊的队列，用于存储由于以下原因无法被正常处理的消息。

a）**消息过期：**消息超过设定的生存时间；

b）**队列已满：**目标队列达到容量限制，无法接收新的消息；

c）**消息被拒绝：**消费者拒绝消息并设置 requeue=false 时，消息将不会重新排入队列。

7.3.5 消息处理策略

在 BACnetMQ 网络中，当出现无法路由的消息时，系统应根据节点策略采取以下处理方式之一：

a）**引导进入备用交换机：**将消息发送至备用交换机，由备用交换机执行指定的路由逻辑，例如重定向到其他队列或特定的处理模块；

b）**直接进入死信队列：**将消息存储到死信队列中，由专门的主机或服务进行后续处理；

c）**直接放弃：**丢弃消息，不做进一步处理。

备用交换机和死信队列中的消息可由专用主机接收，并依据系统需求执行进一步的逻辑处理，如状态维护、问题排查等。

7.3.6 广播包特殊处理

为避免重复消息处理，相同时间戳的广播包若第二次进入同一中心节点的消息队列，应直接丢弃，不再进入死信队列或备用交换机。

7.3.6.1 规范要求

广播包及异常消息的处理应符合下列要求：

a）**备用交换机配置：**系统应根据应用场景合理配置备用交换机，并明确其路由逻辑和目标队列；

b）**死信队列机制：**所有关键队列应具备死信处理能力，以确保消息处理的完整性和可靠性；

c）**广播消息丢弃规则：**广播包应包含唯一标识（如时间戳）。重复广播包应直接丢弃，避免对系统资源的浪费。

7.4 Mesh 网络的路由机制

7.4.1 路由机制概述

在 Mesh 网络中，路由机制应支持多个中心节点的协作，动态维护路由表，并根据实时信息和预定义规则高效地转发消息。每个中心节点负责实时维护一个动态路由表，以指导消息的转发路径，确保消息能够准确到达目标节点。通过这种动态路由机制，Mesh 网络能够提供高效、可靠和可扩展的通信能力。

7.4.2 路由定义

7.4.2.1 规范要求

路由机制应满足以下要求：

a）动态路由表：中心节点应支持动态路由表的创建和维护，路由表内容包括目标节点、下一跳节点和路径权重等信息。

b）广播隔离：路由机制应阻止广播消息的无序扩散，同时支持细粒度的访问控制策略。

c）自修复能力：网络中断时，路由机制应在规定时间内自动修复，并恢复通信链路。

d）负载均衡：路由机制应具备均衡分配负载的能力，避免个别节点过载导致通信中断。

路由是指指导消息从源地址发送到目标地址的路径信息，也可以理解为数据包从源节点传递到目标节点的过程。路由机制的核心包括以下功能：

a）**路径选择：**根据实时网络状态选择最优路径。

b）**动态调整：**在网络拓扑变化或节点状态改变时，路由表能够自动更新。

c）**目标传递：**确保数据准确无误地到达目标节点。

路由机制是Mesh网络的一个核心特征，决定了Mesh网络自组网的效率和可恢复性，应具备以下特征：

a）路由表构建与维护

每个中心节点实时维护一张动态路由表，用于记录从源节点到目标节点的最佳路径。

路由表的更新依据实时网络状态和预定义的规则，确保消息转发的准确性和高效性。

b）隔离广播与访问规则

广播隔离：为避免广播风暴，Mesh 网络应设计机制阻止广播消息的无控制扩散。

访问控制：通过设置访问控制列表（ACL）对流量进行精细化管理，确保合法通信并阻止未授权的消息流。

c）自修复能力

在设备连接断开或节点损坏的情况下，Mesh 网络能够自动更新路由表并重新计算通信路径，确保网络的连通性和稳定性。自修复机制通过动态调整路由表，显著提高网络的可靠性和容错能力。

d）负载均衡

Mesh 网络支持多路径传输，能够在多个节点之间均匀分配网络负载。

负载均衡机制通过避免单一节点过载，提升了整体网络的稳定性和可靠性。

7.5 路由表的建立与维护

7.5.1 路由信息的结构

路由信息应包含以下关键字段，以确保路由表的完整性和实用性：

a）**目标节点ID：**唯一标识目标节点，用于确定消息的接收者。

b）**连接信息：**包含目标节点的IP地址、端口号，以及其他必要的连接参数。

c）**状态：**指示目标节点的当前状态，例如在线、离线、故障等。

d）**最后更新时间：**记录路由信息的最新更新时间，用于判断信息的时效性并触发必要的更新操作。

7.5.2 路由表的创建和维护过程

a）初始化路由表

在系统启动时，每个中心节点应初始化一个空的路由表。

路由表用于存储其他节点的标识（ID）及其对应的连接信息。

b）节点发现与注册

当一个新节点加入系统时，它应向已存在的中心节点注册自己的唯一标识（UUID）及连接信息。

已注册的节点应将新的节点信息广播至路由表中的其他节点，确保全网节点信息一致。

c）动态更新机制

路由表应能够动态反映节点的状态变化。例如：

上线：新增节点后，其他节点应更新路由表以添加新的连接信息。

下线：节点主动退出或断开连接时，其他节点应在路由表中标记其为不可用或移除。

故障：检测到节点异常后，路由表应更新其状态为“故障”。

d）心跳检测与故障恢复

心跳检测：通过定期发送心跳包监测连接节点的状态。如果检测到节点长时间未响应，则应将其状态更新为“离线”或“故障”。

故障恢复：当故障节点恢复连接时，应允许其重新注册，恢复路由信息并同步到其他节点。

7.6 Mesh 网络的路由算法

7.6.1 路由算法概述

Mesh 网络中路由算法的选择直接影响网络通信的效率、可靠性和扩展性。根据智慧建筑控制系统的需求，常用的路由算法包括距离向量路由算法和链路状态路由算法，具体描述如下。

7.6.2 距离向量路由算法

7.6.2.1 算法要求

采用距离向量路由算法时，应实现AODV（Ad-hoc On-demand Distance Vector Routing）算法。

7.6.2.2 工作流程

AODV算法的工作流程应符合下列要求：

a) 路由请求（RREQ）：当源节点需要与目的节点通信时，应向网络广播路由请求消息。该消息应包含源节点地址、目的节点地址及序列号等信息。

b) 路由回复（RREP）：中间节点或目的节点接收到路由请求后，若可提供到达目的节点的有效路径，应返回路由回复消息。

c) 路径建立：当确定到达目的节点的路径后，路由回复消息应沿该路径返回至源节点，以建立通信链路。

7.6.3 链路状态路由算法

7.6.3.1 算法要求

采用链路状态路由算法时，其设计与实现应满足以下要求：

a）资源优化：应兼顾网络资源消耗与计算效率，避免过度的链路状态广播导致网络拥塞或节点负载过高。

b）故障处理：应具备链路状态异常检测和路由快速收敛能力，确保在链路中断时能够迅速重新计算并建立最优路由。

c）扩展性：应支持网络节点的大规模动态加入与退出，确保在高密度设备接入时，路由计算与更新的性能保持稳定。

7.6.3.2 工作流程

链路状态路由算法的工作流程应符合下列步骤：

a) 链路状态更新：每个节点应定期或当链路状态变化时，向网络内泛洪广播链路状态通告，该通告应包含本节点的邻居信息及与邻居间的连接状态（如代价）。

b) 全局视图构建：所有节点通过交换和收集链路状态通告，应构建并维护一个统一的、全局的网络拓扑视图数据库。

c) 最短路径计算：每个节点应基于全局拓扑视图，运行最短路径优先算法（如Dijkstra算法），独立计算到达网络中所有其他节点的最优路径，并生成路由表

7.6.3.3 应用场景

链路状态路由算法主要适用于以下场景：

a) 稳定核心网络：适用于智慧建筑中拓扑结构相对稳定、要求高效且全局路由优化的核心控制网络。

b) 对网络状态感知要求高的场景：适用于需要精确、实时掌握全网拓扑状态以实现精细化管理或高级服务的场景。

7.6.3.4 路由算法选择原则

系统应根据实际网络环境的特点选择合适的路由算法，选择时宜考虑以下因素：

a) **动态性环境**：对于网络拓扑频繁变化的动态环境（如移动设备临时组网），宜优先采用按需驱动的距离向量路由算法（如AODV）。

b) **稳定性环境**：对于拓扑结构稳定的核心网络环境，且要求全局最优路径时，宜采用链路状态路由算法。

7.7 访问控制 ACL

7.7.1 一般要求

网络应实施访问控制列表机制，对数据流进行精细化管理，以防止广播风暴、识别无效数据并过滤潜在恶意流量。

7.7.2 规范要求

ACL的配置与执行应满足以下要求：

a) 广播频率限制：

- 应限制每个节点在特定时间窗口（如1 min）内的广播次数，以防止广播风暴。
- 应设定默认的每分钟广播次数上限。
- 当节点广播频率超出预设阈值时，ACL应自动阻止其后续广播流量。

b) 重复数据包处理：

- 应为所有数据包添加唯一性标记（如时间戳、序列号），以供ACL进行识别。
- 时间戳与序列号应作为数据包的必备字段，用于判定数据包的新鲜度与唯一性。
- ACL应设置机制，对识别出的重复数据包或环路转发数据包立即丢弃。
- 对丢弃的重复或环路数据包，ACL应记录相关异常行为。

c) 流量过滤：

- ACL应支持基于以下条件的精细化流量过滤规则：
- 来源地址：阻止来自未授权节点的数据。
- 目的地址：限制对特定节点或网段的访问。
- 内容特征：识别并阻止不符合协议规范或包含恶意指令的数据包。

8 网络安全机制

8.1 通信协议安全机制

智慧建筑常用的通信协议（如 MQTT 和 AMQP）应基于传输安全协议TLS（Transport Layer Security） 运行，TLS应提供加密、身份验证与完整性校验功能，其实现应符合下列要求：

a) 加密：应对传输数据进行加密，确保通信内容的机密性，防止信息泄露。

b) 身份验证：应在TLS握手阶段通过验证数字证书，确保通信双方身份的合法性。

c) 完整性：应通过消息认证码等校验机制，确保数据在传输过程中未被篡改。

TLS 实现中使用的密码算法应符合国家密码管理相关规定，宜采用 GB/T 32907-2016 和 GB/T 32918 系列标准定义的算法。

8.2 密码算法的适配性

系统中所使用的密码算法，其选型与实现应符合国家及行业标准，并适应具体的部署场景。密码算法应提供以下安全功能：

a) 密钥交换与身份验证：应支持安全的密钥协商机制与双向身份验证。

b) 消息认证与完整性校验：应提供用于检测数据篡改的杂凑或消息认证算法。

c) 数据加密：应提供对传输数据及敏感信息进行加密保护的对称或非对称密码算法。

8.3 国密算法在网络安全中的应用（可选实现）

在涉及智慧建筑控制系统的网络安全中，可根据场景需求选择使用国密算法（SM2、SM3、SM4）作为加密和身份验证机制，以满足国家对信息安全的相关要求。

在涉及国家信息安全要求的智慧建筑控制系统场景中，可根据选择使用国家密码管理局批准的商用密码（SM2、SM3、SM4）算法。

8.3.1 SM2 公钥密码算法应用

SM2算法应用于TLS时的实现应符合下列要求：

a) **功能**：应用于密钥交换与服务端/客户端身份验证。

b) **实现要求**：在TLS握手过程中，应使用符合GB/T 32918.1～32918.5要求的SM2算法进行证书传递与密钥协商。

8.3.2 SM3 杂凑算法应用

SM3算法应用于TLS时的实现应符合下列要求：

a）功能：应用于消息认证与完整性校验。

b）实现要求：在生成报文认证码或进行完整性校验时，宜使用SM3杂凑算法。

SM3 哈希算法的使用应符合 GB/T 32905 相关规定。

8.3.3 SM4 分组密码算法应用

SM4 可作为一种加密算法，用于加密在 TLS 连接上传输的数据。在 TLS 握手过程中，客户端和服务器协商使用 SM4 算法来保护后续的数据传输。SM4 分组密码算法的实现应符合 GB/T 32907-2016 的规定。

SM4算法应用于TLS时的实现应符合下列要求：

a）功能：应用于传输数据的对称加密。

b）实现要求：在 TLS 握手过程中，客户端和服务器协商使用 SM4 算法来保护后续的数据传输，可选择使用符合GB/T 32907-2016要求的SM4算法对通信数据进行加密保护。

8.4 通用安全规范要求

智慧建筑控制系统数据通信的安全设计与实现应满足以下通用要求：

a）标准符合性：密码算法的选择与实现应遵循国家标准和行业规范，确保与主流设备及系统的兼容性。

b）特殊场景支持：对于有特定信息安全要求的应用场景，应支持配置并使用国密算法或其他经认可的行业标准密码算法。

c）开放与可适配性：系统安全架构应具备灵活性，支持密码算法的更新与替换，以适应未来技术发展和特定场景需求。

8.5 典型应用场景

本章规定的安全机制主要适用于但不限于以下场景：

a）数据加密传输：对系统内的关键控制指令、配置参数、用户隐私等敏感信息进行加密传输。

b）设备身份验证：在设备接入网络或进行关键操作前，验证其身份合法性，防止非法设备接入。

c）通信数据完整性保护：对重要的状态数据、告警信息、日志记录等进行完整性校验，防止数据在传输中被恶意篡改。

9 互联互通

9.1 兼容路由 BACnet/IP

为确保BACnetMQ网络能够路由和处理本地BACnet/IP消息，应实现BACnet/IP与BACnetMQ之间的协议转换，具体实现应符合下列要求。

9.1.1 协议转换要求

a）消息格式转换：BACnet/IP 消息应在 BACnetMQ 节点中被转换为 MQTT 消息格式，同时保留 BACnet/IP 的源地址、目标地址及路由信息，以确保通信的准确性和可追溯性。

b）分段协议路由：在混合网络中，根据节点所在网段的协议特性，采用分段路由策略，实现跨协议的无缝通信。

节点 A 与节点 B 通信（同一网段）：A 和 B 直接使用 BACnet/IP 协议进行通信，避免额外的协议转换。

节点 B 与节点 C 通信（跨网段）：B 和 C 使用 BACnetMQ 协议进行通信，确保跨网段通信的效率和可靠性。

节点 C 与节点 A 通信（跨网段与协议转换）：消息需通过节点 B 路由，B 负责将 BACnet/IP 协议消息与 BACnetMQ 协议消息相互转换。

路由过程示例

节点 A → 节点 C：

步骤一：节点 A 发送 BACnet/IP 消息至节点 B。

步骤二：节点 B 将 BACnet/IP 消息转换为 MQTT 消息格式并发送至节点 C。

步骤三：节点 C 接收并解析 MQTT 格式的消息，完成跨网段通信。

节点 C → 节点 A：

步骤一：节点 C 将 MQTT 消息发送至节点 B。

步骤二：节点 B 将 MQTT 消息转换为 BACnet/IP 消息格式并发送至节点 A。

步骤三：节点 A 接收并解析 BACnet/IP 消息，完成双向通信。

9.1.2 规范要求

兼容路由BACnet/IP 应满足以下规范要求：

a）协议转换支持

BACnetMQ 节点应支持 BACnet/IP 消息的接收、转换和转发能力。

转换过程中应保留原始地址信息（src/dst）和路由信息，确保通信链路的完整性。

b）分段通信策略

在同一网段中，节点之间优先使用原生 BACnet/IP 协议通信。

跨网段通信时，应使用 BACnetMQ 协议，并通过边界节点（如节点 B）实现协议转换。

c）路由效率

BACnetMQ 节点应具备高效的路由能力，确保协议转换过程不会显著增加网络延迟。
路由机制应支持动态更新，适应网络拓扑变化。

D)兼容性
BACnetMQ 节点应兼容现有 BACnet/IP 设备，确保新旧系统的互联互通。
在协议转换过程中，应保持 BACnet/IP 的功能特性不受影响。

9.2 兼容路由 BACnet/MSTP

9.2.1 协议转换要求

为了实现 BACnetMQ 网络对本地 BACnet/MSTP 消息的路由，应将 BACnet/MSTP 的通信数据转换为 MQTT 消息格式，并保留 BACnet/MSTP 相关的 BVLC 地址信息与路由信息。在无 IP 转发需求的情况下，可在消息的 src/dst 区段中填充本地 IP 或其他特征码进行标识。

具体的兼容路由要求如下：

a) 路由场景描述
节点 A 与节点 B 通信（同一总线）：
节点 A 和节点 B 位于同一 BACnet/MSTP 总线，直接使用 BACnet/MSTP 协议通信。
节点 B 与节点 C 通信（跨网段）：
节点 B 和节点 C 位于不同网段，节点 B 将 BACnet/MSTP 消息转换为 MQTT 消息格式，使用 BACnetMQ 协议与节点 C 通信。
节点 C 与节点 A 通信（跨网段与协议转换）：
在节点 C 与节点 A 通信时，消息应通过节点 B 路由，节点 B 负责在 BACnet/MSTP 协议和 BACnetMQ 协议之间进行转换。

b) 路由过程示例
节点 A → 节点 C：
① 节点 A 将 BACnet/MSTP 消息发送至节点 B。
② 节点 B 将 BACnet/MSTP 消息转换为 MQTT 消息格式，并将消息发送至节点 C。
③ 节点 C 接收并解析 MQTT 格式消息，完成跨网段通信。

节点 C → 节点 A：
① 节点 C 将 MQTT 消息发送至节点 B。
② 节点 B 将 MQTT 消息转换为 BACnet/MSTP 消息格式，并将消息发送至节点 A。
③ 节点 A 接收并解析 BACnet/MSTP 消息，完成双向通信。

9.2.2 规范要求

a) 协议转换支持
BACnetMQ 节点应支持 BACnet/MSTP 消息的接收、转换和转发能力。
转换过程中应保留原始 BVLC 地址信息和路由信息，确保消息的完整性和可追溯性。
b) src/dst 填充规则
在无 IP 转发需求时，src/dst 字段应填充本地 IP 或唯一特征码，用于标识消息的路由来源和目标。
c) 分段通信策略
BACnet/MSTP 节点在同一总线内直接使用 BACnet/MSTP 协议通信。
跨网段通信应通过兼容 BACnet/MSTP 的 BACnetMQ 节点完成协议转换。
d) 路由效率
BACnetMQ 节点的路由机制应确保协议转换高效完成，避免增加网络延迟。
路由表应动态更新，以适应网络拓扑变化。

e) **兼容性**

BACnetMQ 节点应与现有 BACnet/MSTP 设备兼容，确保系统升级或扩展时无缝集成。

9.3 BACnet 数据的封装

BACnet 数据封装结构（包括BVLC、NPDU、APDU）应符合 GB/T 28847.5 中要求。如图8所示，BACnet 数据在 BACnetMQ 网络中的封装格式由多个关键字段组成，涵盖了通信节点标识、网络层地址信息及应用层协议单元。各字段有对应的作用和定义。

a) **封装结构标准化**

BACnetMQ 节点应按照上述封装格式处理数据包，确保数据的可互操作性。

UUID、src/dst 地址字段应清晰定义，保证消息的追溯性和路由准确性。

b) **兼容性支持**

BVLC、NPDU 和 APDU 的格式应完全兼容 BACnet/IP 的既有规范，确保与现有 BACnet/IP 系统的无缝集成。

c) **消息完整性与安全性**

数据封装应支持消息完整性校验，确保在传输过程中数据不会被篡改。

适配网络安全机制（如 TLS），对封装的消息进行加密和保护。

d) **灵活性与扩展性**

结构设计应支持 BACnetMQ 的扩展功能（如新字段或自定义应用），同时兼容标准化字段的核心功能

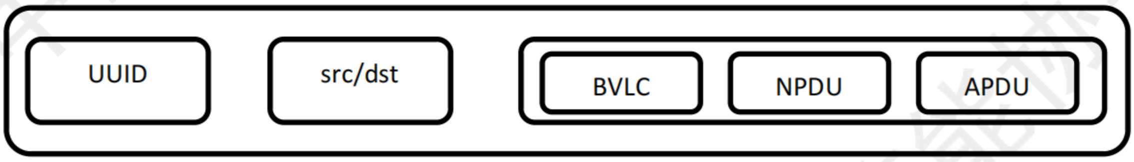

图 8 BACnet数据包在BACnetMQ网络上的消息封装

9.3.1 UUID

描述：表示 BACnet/IP 接入的 BACnetMQ 节点的唯一标识符。
作用：用于在 BACnetMQ 网络中唯一标识消息的发送或接收节点，确保网络中不同节点的识别与追溯。

9.3.2 src/dst IP+Port

描述：消息发起方和接收方的 IP 地址与端口号。
作用：记录消息的源地址（src）和目标地址（dst），便于消息的路由与定位。若无 IP 转发需求，此字段可填充本地 IP 或特征码以标识。

9.3.3 BVLC (BACnet Virtual Link Control)

描述：BACnet 虚拟链路控制字段，沿用 BACnet/IP 的 BVLC 格式。
作用：提供虚拟链路层控制功能，支持消息的广播和路由等操作。

9.3.4 NPDU（Network Protocol Data Unit）

描述：网络协议数据单元，用于承载网络层的信息，沿用 BACnet/IP 的 NPDU 格式。
作用：包含网络层的目标信息、路由信息和控制标志，为数据的跨网络传输提供支持。

9.3.5 APDU（Application Protocol Data Unit）

描述：应用协议数据单元，用于包含 BACnet 应用层的信息，沿用 BACnet/IP 的 APDU 格式。
作用：承载应用层数据，支持 BACnet 应用协议功能，如读取属性、控制设备等。