白皮书

通过MQTT的设备集成

三月 2022

AXIS COMMUNICATIONS

概述

MQTT是一款标准消息协议，有助于在IoT设备与云应用中间高效可靠地交换数据。它让设备能够（通过其MQTT客户端）将消息发布到公共MQTT Broker（服务器），此Broker能够协调与其他设备的通信。此Broker记录谁发布了什么以及谁希望查看这些数据，进而将消息仅转发到订阅了相应主题的客户端。

在典型VMS生态系统中，来自设备的安讯士事件通知以往使用RTSP流传输协议通过VAPIX/ONVIF API接口流送到单一目的地。但这些事件通知现在可以通过MQTT协议通过安讯士设备内置的MQTT客户端（适用于运行AXIS OS 9.80或更新版本的设备）进行发布。这对于VMS生态系统的内部和外部都适用，且尤其适用于互联网环境。然后，网络上的多个被订阅的MQTT客户端能够使用并处理安讯士设备所发布的事件通知。某些安讯士和第三方ACAP分析应用也具有自己的MQTT客户端，这些客户端专为特定的系统、应用场合和订阅者设计。

以涉及安讯士产品的一个应用场合为例，人数统计设备能够通过MQTT将统计数据发送到云中的数据可视化软件。又如，第三方门传感器通过MQTT与报警设备和摄像机通信，每次开门时，报警设备和摄像机便会发出报警并启动录像。

目录

1 引言 4

2 MQTT协议 4

3 优点 5

4 缺点 5

5 基础设施 6

6 安全 6

7 安讯士设备中的MQTT客户端 6

8 ACAP分析应用中的MQTT客户端 7

9 其他MQTT客户端 7

10 通过MQTT的设备集成应用示例 8

10.1 传送到云平台面板的人数统计分析数据 8

10.2 通过MQTT获取的门传感器数据可触发报警设备的报警以及摄像机的录像 9

11 词汇表 10

12 商标归属说明 11

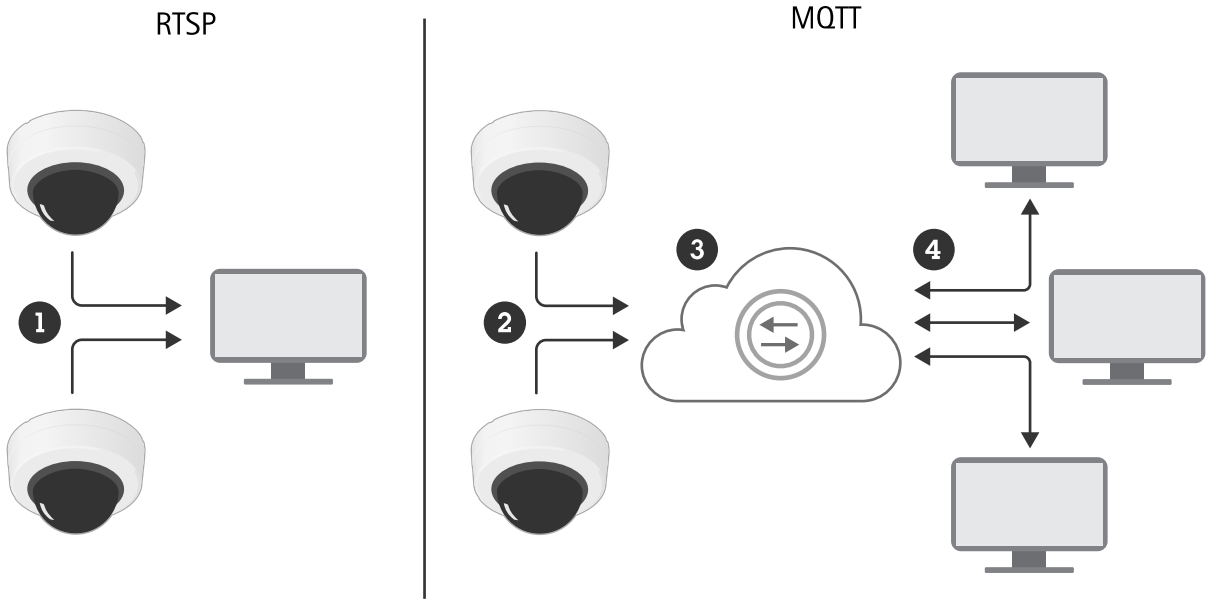
1 引言

MQTT（消息队列遥测传输）是用于物联网 (IoT) 的标准消息协议。它旨在简化IoT集成，并在不同行业中使用，以较小的代码需求量和尽可能小的网络带宽远程连接设备。AXIS OS中的MQTT客户端可使设备中的数据和事件集成至非视频管理系统系统 (VMS) 的流程简化。

本白皮书介绍了MQTT的技术背景，包括其典型应用场合及优缺点。其中还详细介绍了安讯士设备和ACAP分析应用中的MQTT客户端。

2 MQTT协议

MQTT是一种发布/订阅协议。这就意味着，它的消息传输模式与作为请求/响应协议的RTSP或HTTP不同。在使用RTSP的情况下，一方发出请求，另一方做出响应。许多移动消息应用都改为了使用MQTT或类似的发布/订阅概念。也有的发布/订阅协议在经过优化后可适用于封闭系统或特定的系统。



RTSP仅允许一对一通信，但MQTT还支持通过Broker的一对多或多对一通信。

1 *事件流*
2 *发布*
3 *MQTT Broker*
4 *订阅*

MQTT概念是，所有客户端连接到公共MQTT Broker（服务器），而此Broker则记录谁发布了什么以及谁希望查看这些数据。连接通常是TCP会话的形式，通过端口1883建立。客户端也可以通过TLS（通常是8883）建立，或者也可以使用WebSocket（通常是端口1884/8884）来建立。

客户端发布包含主题的消息。别的客户端可以订阅此主题，或者可以使用通配符获取所有副主题。消息还包含通常为JSON数据结构、字符串或甚至短二进制数据的有效载荷数。发送者不知道其他客户端是否正在订阅。Broker只会将消息转发到具有相应主题订阅的客户端。

在使用MQTT的情况下，其工作方式有点类似于将文章发表到杂志中。订阅了杂志的人将能够阅读此文章，它可以是一对一或一对多通信（MQTT甚至还支持多对多通信）。也可以在距离首次发布很久之后才阅读此文章。

相比之下，RTSP的使用更类似于打电话。命令仅涉及一个来源和一个目标，且始终是一对一通信。如果目标未接电话，就会错过消息。

当使用MQTT发布来自设备的安讯士事件通知时，网络中的多个被订阅的MQTT客户端可以使用并处理这些通知。相比将事件通知流送到仅一个目的地的传统方式（使用 VAPIX®/ONVIF® 应用编程接口 (API) 和 RTSP），这是一大优势。

3 优点

MQTT有许多优点。较之于使用请求/响应协议（如RTSP），MQTT的主要优点表现为：

- **降低了设备密码的暴露风险。** 客户端在获取数据时，不需要访问设备或服务器。这就意味着，客户端不需要知道密码，也不需要知道API的工作方式。这就降低了设备密码暴露于客户端和用户的风险，从而降低了刻意或意外误用的风险。
- **单点集成。** 在经授权的情况下，所有客户端都可以通过到Broker的单点连接，获取所有其他客户端的已发布消息。在RTSP中，客户端需要连接到要从中获取数据的各个其他客户端。这就意味着，MQTT消息流可能是一对一、一对多、或者多对一，每个客户端不会承受额外的负担。
- **在遵守防火墙规则的前提下发布和订阅**在RTSP中，客户端需要能够访问设备/服务器API。如果设备位于防火墙后方且客户端为远程客户端，则需要将防火墙配置为允许入站请求，从而暴露设备API。在中间部署有公共MQTT Broker的情况下，防火墙后方的客户端可以在不破坏防火墙规则的前提下发布/订阅特定数据（如果防火墙允许出站连接）。
- **服务质量。** 在发布关键消息时，发布者可以监视该消息是否被别的客户端接收，如果答案为否，则会采取替代措施。
- **留存消息。** 发布者可以将消息标记为“留存”，这就意味着Broker将保留消息副本，并将此消息发送至新连接且订阅了该主题的客户端。
- **IoT客户端可用性。**MQTT客户端包在大多数常用软件开发环境（包括Windows®、Linux®、Android™、iOS、Node.js®、PHP和Python®）中都可用。较之于建立到设备的RTSP数据流，可以将更多的客户端连接到Broker。
- **简化消息监视与除错。** 有多款MQTT工具都可以用来监视所发布的消息，以及用来发布消息以便对订阅者是否响应和如何响应进行故障排查。
- **简化数据结构。** 由于MQTT通常以未知客户端为目标，因此消息的有效载荷数据将通常会考虑这一点，以便简化订阅者的使用。

4 缺点

较之于其他协议，MQTT存在一些缺点：

- **单点故障。** 如果Broker不可用，则所有传输将停止。但可以在基础设施中设计冗余Broker，以解决这一问题。

- **消息由谁发布?** 在设计上，MQTT聚焦于主题而不是消息由谁发布。除非发布者在主题或有效载荷数据中包含了某个ID，否则将需要访问Broker的日志，才能知道消息是由谁发布的。通常的做法是，发布者根据具体的应用情形在主题或有效载荷数据中包含某个客户端ID。
- 连接到Broker的**恶意客户端**可能发布/订阅自身权限范围内的任何主题。因此需要保护Broker（参见有关MQTT安全的章节）。
- 它**不适用于连续流送视频/音频。**

跟其他服务器一样，它也需要考虑总的带宽吞吐量。对于包含众多客户端的规模非常大的系统，可能需要执行动态扩展。

5 基础设施

本地Eclipse Mosquitto™ Broker的建立非常轻松，或者也可以非常轻松地使用Node-RED®（如Aedes）充当本地Broker。也有许多互联网服务提供商以及其他机构提供了管理型MQTT Broker，比如Microsoft® Azure® IoT、HiveMQ™、CloudMQTT和IBM® Cloud®。

如果系统没有远程客户端，则建议使用本地Broker。本地Broker还能够充当公共Broker的代理，或者也可以被配置用于充当特定本地Broker消息和公共Broker消息的代理。

6 安全

Broker需要根据消息的关键性以及具体系统可能面临的威胁获得适当的保护。MQTT提供了若干不同的身份验证方案，其中包括无验证、用户/密码以及TLS客户端证书验证。在能够发布或订阅的主题方面，不同的用户可能需要使用不同的身份验证方案。Broker能够允许客户端通过未加密的TCP或者通过加密的TLS（如HTTPS）进行连接。

- **无验证：** 如果消息是非关键消息且本地Broker不会暴露给互联网客户端，则该Broker可以禁用身份验证。建议仅在测试、沙盒开发和演示中使用这种身份验证方案。
- **用户/密码：** 这是较常用的身份验证方案。系统管理员可以根据系统的潜在风险让所有MQTT客户端共享相同的用户/密码，也可以创建拥有受限主题访问权限的用户。
- **TLS客户端证书：** 对于暴露在互联网环境中的Broker，如果其中的消息属于敏感消息，则应将此Broker配置成仅允许具有有效TLS证书的客户端。这种方案需要用到PKI（公钥基础设施），并需要有能够签发受Broker信任的客户端证书的证书授权机构。公共MQTT互联网服务提供商通常都能够做到这一点。

在某些情况下，可能适合以多个Broker（本地和/或公共Broker）来划分不同的应用情形。划分关键消息和非关键消息是一种安全控制措施。多个Broker也将降低单点故障风险，并有助于增强监控和故障排查。其代价在于，需要对额外的Broker进行额外的开发和维护。

7 安讯士设备中的MQTT客户端

在标准VMS生态系统中，来自设备的安讯士事件通知以往使用RTSP流传输协议通过VAPIX/ONVIF API接口流送到单一目的地。

这些事件通知现在可以通过MQTT协议通过安讯士设备（其运行AXIS OS 9.80或更新的版本）内置的MQTT客户端进行发布。这对于VMS生态系统的内部和外部都适用，且尤其适用于互联网环境。利用MQTT，网络中的多个被订阅的MQTT客户端能够使用并处理安讯士设备所发布的事件通知。

8 ACAP分析应用中的MQTT客户端

某些安讯士和第三方ACAP应用具有自己的MQTT客户端，这些客户端专为特定的系统、应用场合和订阅者设计。Axis Publisher就是其中一个例子，它添加了某些系统所需的额外特性、结构和行为。

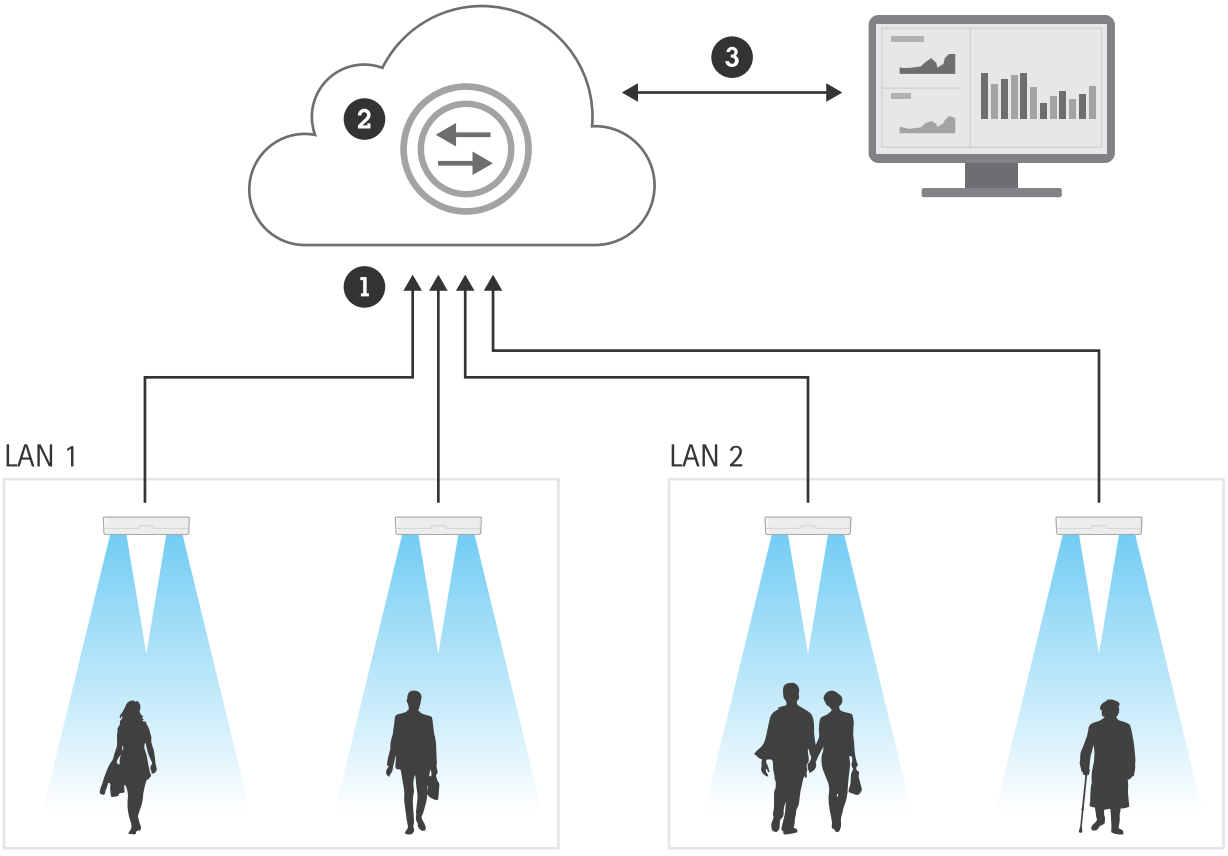
9 其他MQTT客户端

有许多MQTT客户端都可以安装在Linux、Windows、Android和iOS系统上，这些客户端被设计成工具或服务，用于特定应用。MQTT非常适用于脚本处理以及Node–RED/Node.js、Python、PHP等中间件。大多数互联网服务平台（如Microsoft Azure IoT、AWS™和Google Cloud Platform™）都提供了MQTT Broker以供集成到在平台上运行的服务中。大多数传感器、移动应用和（家用）自动化系统都具有MQTT客户端。

10 通过MQTT的设备集成应用示例

10.1传送到云平台面板的人数统计分析数据

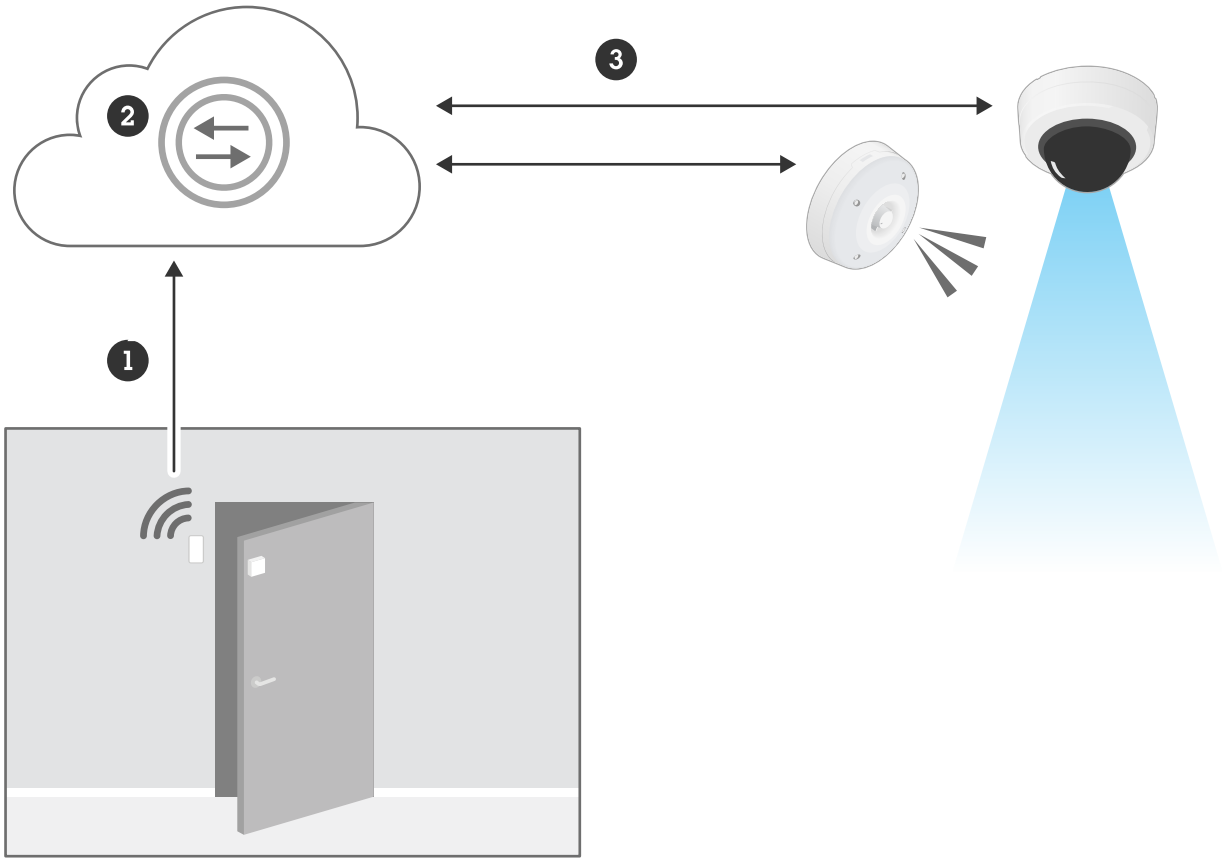
每当检测到有人“进/出”预定义区域时，搭载有人数统计分析工具的设备便会生成事件通知。通知被传送到MQTT客户端，后者则将其实时发布到云平台。在云平台中，建立了到数据可视化软件（比如Microsoft® Power BI®面板）的连接，以便显示来自人数统计器的实时统计数据。



1 发布
2 MQTT Broker
3 订阅

10.2通过MQTT获取的门传感器数据可触发报警设备的报警以及摄像机的录像

第三方MQTT门传感器用于触发开门时的事件通知。门传感器将MQTT消息发布到云中的MQTT Broker。报警设备和摄像机订阅门传感器的主题，并在开门时发出报警并启动录像。



1 发布
2 MQTT Broker
3 订阅

11 词汇表

ACAP	*AXIS Camera Application Platform*，一种应用框架，用于在前端添加功能性和智能性
Aedes	一种MQTT Broker
API	*应用编程接口*，一种代码，允许两个软件程序彼此通信
AWS™	一种云服务平台
AXIS OS	前端安讯士设备的操作系统
CloudMQTT	一种MQTT Broker
Eclipse Mosquitto™	一种开源消息Broker，用于实现MQTT协议
Google Cloud Platform™	一种云服务平台
HiveMQ™	一种MQTT Broker
HTTP	*超文本传输协议*，万维网上使用的一种数据传输协议
IBM Cloud®	一种云服务平台
IoT	*物联网*，即，通过互联网对嵌入在日常设备和设施中的计算设备进行互联
JSON	*JavaScript对象表示法*，一种紧凑文件格式和数据交换格式
Microsoft® Azure® IoT	一种云服务平台
Microsoft® Power BI®	一种专注于商业智能的交互式数据可视化软件程序
MQTT	*消息队列遥测传输*，一种用于物联网的消息协议。
Node.js®	一种开源开发平台，用于在服务器侧执行JavaScript代码
Node–RED®	一种用于物联网连接的编程工具
ONVIF®	一个开放式行业论坛，为基于IP的物理安防产品的高效互操作提供和推广标准化接口
PHP	一种面向网页开发的通用脚本语言
Python®	一种通用编程语言
RTSP	*实时流传输协议*，一种网络协议，用于建立和控制端点之间的媒体会话
TCP	*传输控制协议*，一种数据传输协议，是主流互联网协议之一
TLS	*传输层安全*，此协议用于保障计算机网络通信的机密性和完整性
VAPIX®	适用于安讯士产品的开放式应用编程接口 (API)
WebSocket	一种通信协议，通过单一TCP连接提供双向通信
VMS	*视频管理软件*或*视频管理系统*

12 商标归属说明

Android和Google Cloud Platform是Google LLC的商标。

AWS是Amazon.com, Inc.或其附属公司在美国和/或其他国家和地区的商标。

Eclipse Mosquitto是Eclipse Foundation, Inc的商标。

HiveMQ是HiveMQ GmbH的商标。

IBM和IBM Cloud是International Business Machines Corp在全球多个国家和地区注册的商标。

IOS是Cisco Systems, Inc和/或其附属公司在美国及某些其他国家和地区的注册商标，它在Apple, Inc的授权下使用。

JavaScript是Oracle Corporation在美国的注册商标。

Linux是Linus Torvalds在美国以及其他国家和地区的注册商标。

Microsoft、Windows、Microsoft Azure IoT和Microsoft Power BI是Microsoft Corporation的注册商标。

Node.js和Node–RED是OpenJS Foundation在美国和/或其他国家和地区的注册商标。

ONVIF是Onvif, Inc的商标。

Python是Python Software Foundation的注册商标。

T10162730/zh/M1.17/2203

关于 Axis Communications

Axis 通过打造解决方案，不断提供改善 以提高安全性和业务绩效。 作为网络技术 公司和行业领导者，Axis 提供视频监控解决方案， 访问控制、对讲以及音频系统的相关产品和服务。 并通过 智能分析应用实现增强，通过高品质培训提供支持。

Axis 在 50 多个国家/地区拥有约 4,000 名敬业的员工 并与全球的技术和系统集成合作伙伴合作 为客户带来解决方案。 Axis 成立于 1984 年，总部 在瑞典隆德