December 2022

Self-Organizing Publish/Subscribe on the Network Edge

Phil Genera

Self-Organizing Publish/Subscribe on the Network Edge

ABSTRACT

In a publish/subscribe protocol such as MQTT, Internet-of-Things (IoT) devices communicate as follows. A device with a message pertaining to a certain topic publishes the message to a cloud-based server. A device that subscribes to the topic receives messages pertaining to the topic from the cloud-based server. The traditional publish/subscribe protocol fails if the internet is unavailable. This disclosure describes techniques for publish/subscribe messaging without the need for a central coordinating server. A subscriber translates the topic of interest to a multicast IP address. To publish a message, the publisher simply sends a UDP packet with the message content to that address, and it is delivered to all listening subscribers. Inter-device communication is thereby made robust to internet connectivity failures.

KEYWORDS

- Internet of things (IoT)
- Publish-subscribe
- Message queue telemetry transport (MQTT)
- Domain name service (DNS)
- Multicast DNS (mDNS)
- Self-organizing network

BACKGROUND

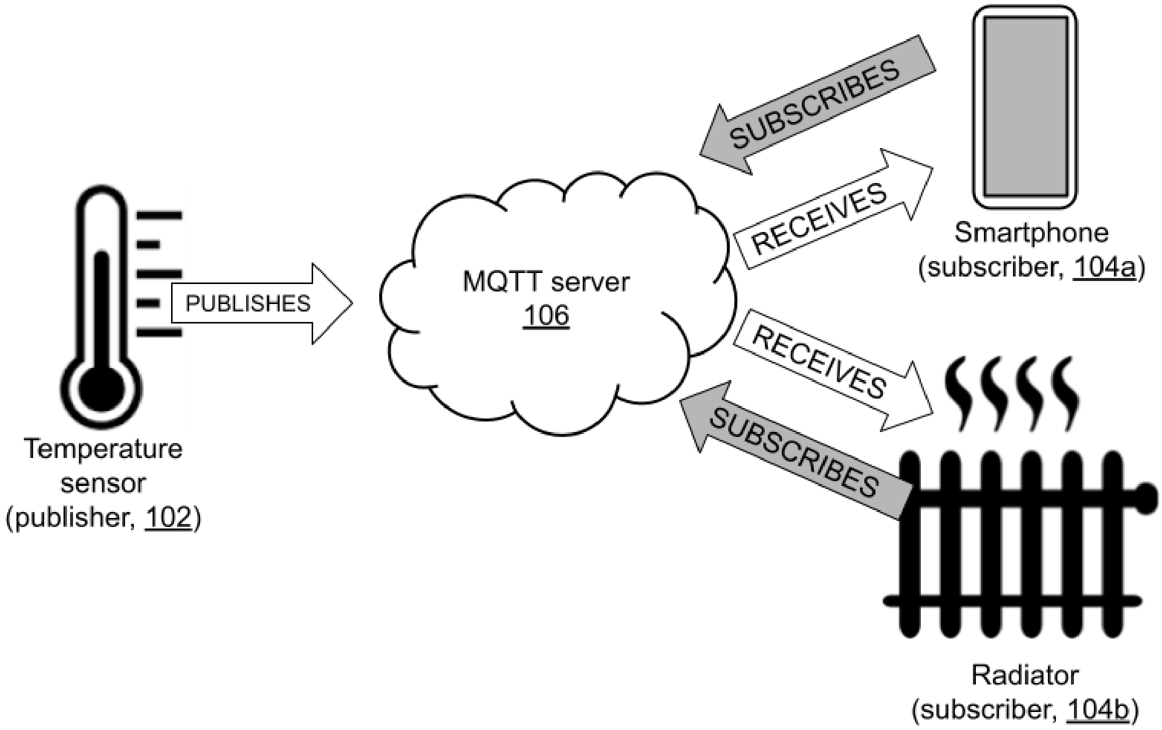

Fig. 1: Traditional MQTT messaging

Internet of Things (IoT) devices often coordinate within the same network broadcast domain. This is generally accomplished using a cloud publish/subscribe server: device A registers for (subscribes to) messages of a given topic, and device B publishes messages to that topic. For example, as illustrated in Fig. 1, a temperature sensor acts as publisher (102) for a certain topic (named ‘temperature’) and publishes messages relating to the topic to the MQTT server (106) located in the cloud. Various subscribers (104a-b), e.g., a smartphone, a radiator, etc., that subscribe to the topic ‘temperature’ receive published messages via the MQTT server. This works well when a connection to the cloud is available, but fails when access to the internet is unavailable or when the server is down.

Although the MQTT protocol provides a client/server implementation of publish/subscribe messaging, it requires a central coordinating authority in the form of an

MQTT server, and is thus vulnerable to internet connectivity failures.

Example

A robot vacuum and a robot mop operate in the same room. The robot mop is to be notified when the robot vacuum completes its job, so that it can mop without risking interference from the robot vacuum. Currently, the vacuum-to-mop communication is implemented by the vacuum publishing mission status to a cloud-based MQTT server; the mop listening for the mission status of the vacuum; and the mop starting its own mission when it sees the task completion status of the vacuum. If the internet connection or cloud server is down, the vacuum-mop coordination fails.

DESCRIPTION

This disclosure describes techniques for publish/subscribe messaging without the need for a central coordinating server. Inter-device communication is thereby made robust to internet connectivity failures.

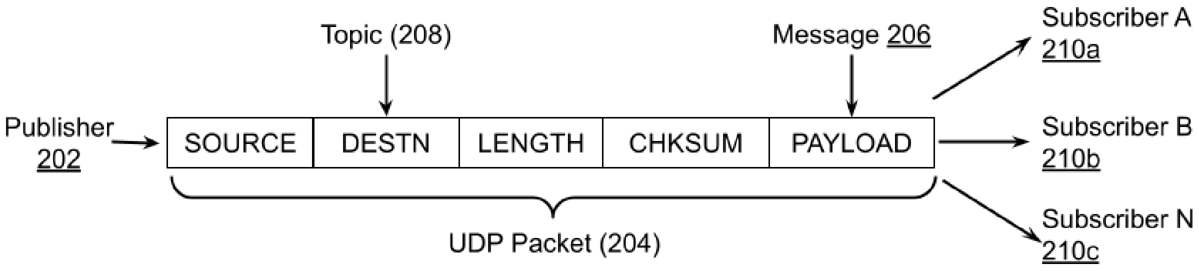

Fig. 2: Self-organizing publish/subscribe on the network edge

Fig. 2 illustrates self-organizing publish/subscribe on the network edge. In lieu of registering with a centralized MQTT server (as is conventional), a subscriber translates the topic of interest to a multicast IP address. For example, for IPv6 networks, the address can be a 112-bit hash of the topic embedded in the lower 112 bits of an appropriately scoped IPv6 address. To publish a message, a publisher (202) simply sends a UDP packet (204) with the message content (206) to that address (208), and the message is delivered to all listening subscribers (210a-c).

In IPv4 networks, or when IP address manipulation isn't desired, the resolution of the multicast address to listen to can be implemented using multicast DNS: A subscriber performs a DNS lookup for a local address with the topic encoded in it, and the publisher responds with an arbitrarily chosen multicast address. The protocol then proceeds as described above.

Example

A robot vacuum and a robot mop operate in the same room. The robot mop is to be notified when the robot vacuum completes its job so that it can mop without risking interference from the robot vacuum. Per the described protocol, the vacuum and mop communicate completely locally without any prior knowledge of each other and convey mission status to each other without the need for a cloud-based server or coordinating authority. The floor gets cleaned even if the internet connection is absent.

In the event there are multiple publishers for a given topic, they all publish to the same address. In mDNS-based coordination, a new publisher performs a DNS resolution to check if there is already a chosen address for a given topic, and use it if there is one. Hash collisions can be mitigated by including a cyclic redundancy check (CRC) of the topic in the published message.

In this manner, the techniques of this disclosure provide an MQTT-like service and API without requiring a central coordinating authority in the form of an MQTT server. Inter-device communication is made robust to internet connectivity failures.

CONCLUSION

This disclosure describes techniques for publish/subscribe messaging without the need for a central coordinating server. A subscriber translates the topic of interest to a multicast IP address. To publish a message, the publisher simply sends a UDP packet with the message content to that address, and it is delivered to all listening subscribers. Inter-device communication is thereby made robust to internet connectivity failures.