From Cloud to Edge: Collaborate MQTT Brokers as a Service

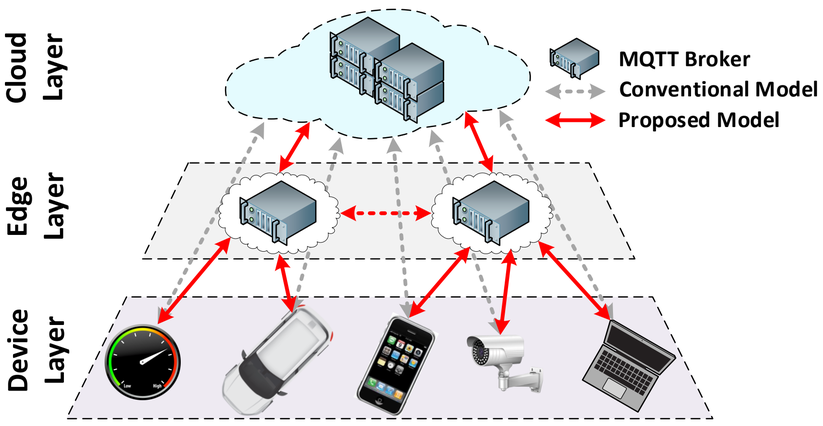
Yanwen Jing, Luoyao Hao, and Henning Schulzrinne
Internet Real-Time Laboratory, Columbia University
Email: yj2556@columbia.edu, {lyhao, hgs}@cs.columbia.edu

Introduction

MQTT: Message Queuing Telemetry Transport
- A lightweight messaging protocol
- Publish-Subscribe (broker, publisher & subscriber)

IoT: Internet of Things
- More and more intelligent devices
- Cloud model: less efficient for high-locality IoT requests

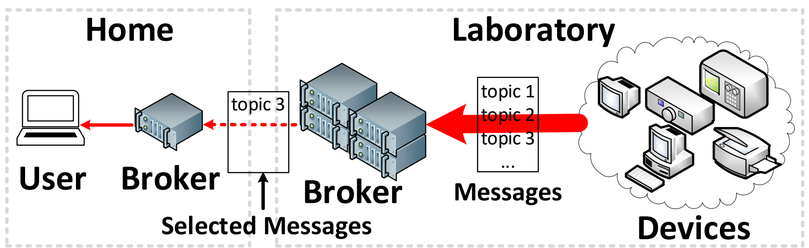
From Cloud to Edge[1]
A central broker on the cloud → Deploy brokers at edges



Moving MQTT brokers to distributed edges

Motivation and Goal

Motivation: Collaborate edge brokers
No missing and no excessive messages



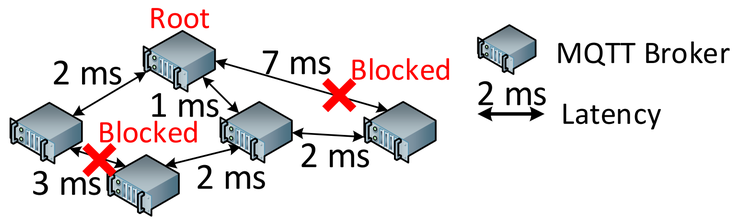
Goal: Accurate and efficient message dissemination

Approach

1. Propose an edge-assisted broker collaboration model;
2. Put a series of algorithms(loop-free, leader election, and dissemination) into practice for edge brokers;
3. Validate our approach through extensive evaluations.

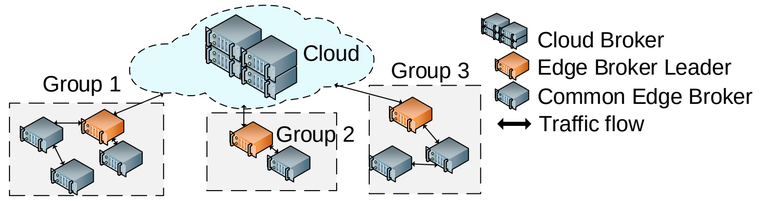
Topology

Loop-Free
Spanning Tree Protocol[2,3] mechanism



Goal: to avoid "subscription storm"

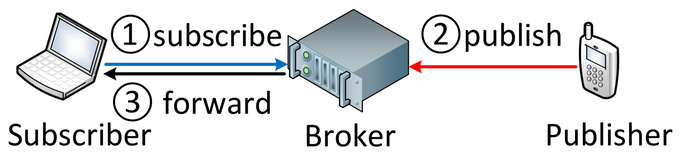
Groups and Leaders



- Brokers in each group maintain loop-free topology
- Elect a broker leader of each edge broker group
- Cloud broker only communicates with leaders

Dissemination Algorithms:

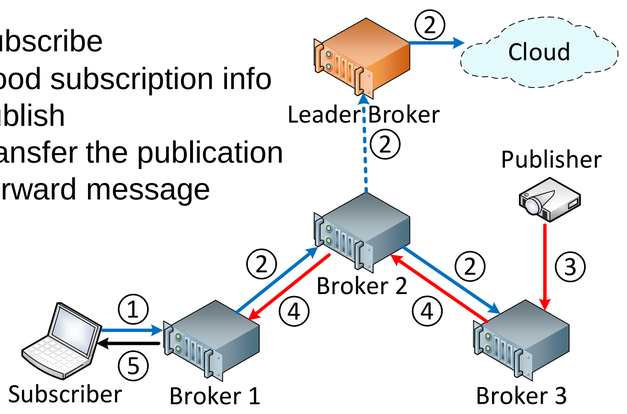
Traditional One-to-One Model



First Option: Subscription Flooding

Subscription Flooding for Multiple Brokers

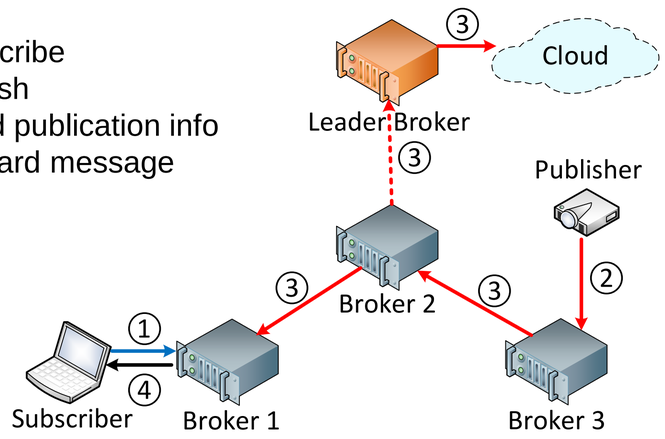
1. Subscribe
2. Flood subscription info
3. Publish
4. Transfer the publication
5. Forward message



Second Option: Publication Flooding

Publication flooding for Multiple Brokers

1. Subscribe
2. Publish
3. Flood publication info
4. Forward message



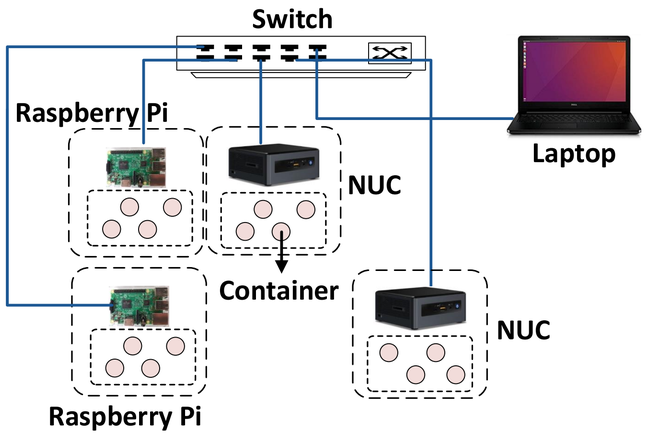
Algorithm Analysis

Subscription flooding
- No redundant data
- Bandwidth saving
- Robustness
- Feasible for intra-group communications

Publication flooding
- Cloud broadcast
- Explicitly announce
- Less storage usage
- Feasible for network states update

Initial Testbed

Laptop: Cloud; Docker containers: MQTT brokers and clients;
Intel NUCs and Raspberry Pis: groups of edge brokers.



Expected results: high throughput, low intra-group latency, fault tolerance, less redundant message

References

[1] Banno et al. "Dissemination of edge-heavy data on heterogeneous MQTT brokers." *IEEE International Conference on Cloud Networking (CloudNet)*, 2017, pp. 1-7.
[2] Sanjuan et al. "Message Queuing Telemetry Transport (MQTT) Security: A Cryptographic Smart Card Approach." IEEE Access 2020, vol. 8, pp.115051-115062.
[3] Longo et al. "MQTT-ST: A spanning tree protocol for distributed MQTT brokers." IEEE International Conference on Communications (ICC), 2020, pp.1-6.
[4] Detti et al. "Sub-Linear Scalability of MQTT Clusters in Topic-Based Publish-Subscribe Applications." IEEE Transactions on Network and Service Management (TNSM), 2020, vol 17, no. 3, pp. 1954-1968.