Poster: An Edge-facilitated Message Broker for Scalable Device Discovery

Zhe Huang
AT&T Labs Research
zhehuang@research.att.com

Bharath Balasubramanian
AT&T Labs Research
bharathb@research.att.com

Azzam Alsudais
University of Colorado Boulder
azzam.alsudais@colorado.edu

Kaustubh Joshi
AT&T Labs Research
kaustubh@research.att.com

CCS CONCEPTS

• **Networks → Naming and addressing**;

ACM Reference format:
Zhe Huang, Bharath Balasubramanian, Azzam Alsudais, and Kaustubh Joshi. 2017. Poster: An Edge-facilitated Message Broker for Scalable Device Discovery. In *Proceedings of SEC '17, San Jose, CA, USA, October 12–14, 2017,* 2 pages.
https://doi.org/10.1145/3132211.3132456

1 ABSTRACT

Searching for a particular device in an ocean of devices is a perfect illustration of the idiom 'searching a needle in a haystack'. Yet the future IoT and edge computing platforms are facing an even more challenging problem because their mission-critical operations (e.g., application orchestration, device and application telemetry, inventory management) depend on their capability of identifying nodes of interest from potentially millions of service providers across the globe according to highly dynamic attributes such as geo-location information, bandwidth availability, real-time workload and so on. For example, a vehicular-based crowd sensing application that collects air quality data near an exit of a highway needs to locate cars in close proximity to the exit among millions of cars running on the road. In a business model where an enterprise offers a framework for clients to avail such edge/IoT services, we investigate the following problem: "among millions of IoT/Edge nodes, how do we locate and communicate with only those nodes that satisfy certain attributes, especially when some of these attributes change rapidly?"

In this paper, we address this problem through the design of a scalable message broker based on the following novel intuition: device discovery should be a joint effort between a centrally managed enterprise-level system (high availability, low accuracy) and the fully decentralized edge (high accuracy, unpredictable availability). To elaborate, the enterprise can centrally maintain and manage the attributes of all the IoT devices. However, since millions of devices cannot constantly update their attribute information, central management has the issue of attribute staleness. Clearly the devices themselves have the most up-to-date information. However, it is not feasible for every request to be routed to million devices connected by unpredictable networks, where only some of them may possess the correct attributes. In this paper, we propose a message broker, in which requests for relatively static device attributes are handled by the centrally managed system, whereas, requests for dynamic attributes are handled by peer-to-peer networks of the edge devices containing those attributes. This combination provides a scalable solution wherein, based on client needs, we can obtain attribute values without compromising on freshness or performance.

There exist several previous works that aim to tackle the device searching problem. Name-based networking solutions such as Intentional Naming System (INS) [1], Auspice [5], and global name service [3] propose to implement a centrally managed name resolution service. Devices periodically update their status information and descriptions in a push approach. While maintaining complete knowledge of every device in the network centrally makes the searching much easier, the excessive workload from millions of devices updating their status in a highly dynamic environment renders the scheme unsaleable. At the other end of the spectrum, a pull-based solution such as Geocast [2] eliminates the status update workload entirely by forwarding device searching query to the devices and relying on the devices to voluntarily identify themselves if their status and attributes match the query. However, the pull-based solutions require an attribute-aware message routing scheme such as distributed hash table [4] that knows exactly how to reach to devices that may match the query. Such design also suffers from longer query response delay caused by query forwarding, and increased security risks because they trust the devices to honestly report their identities and attributes. A better solution should be able to combine the strengths from both the push and pull design principles.

Based on the targeted edge/IoT enviroment and applications, we identify the following design goals for a message broker being able to support a large scale, highly dynamic network enviroment:

- **Searchability.** The message broker must be able to identify and access devices according to arbitrary attributes, service descriptions and queries.
- **Verifiability.** The message broker must be able to verify attributes and descriptions with the authoritative information source.

SEC '17, October 12–14, 2017, San Jose, CA, USA

ACM ISBN 978-1-4503-5087-7/17/10…$15.00
https://doi.org/10.1145/3132211.3132456

- **Scalability.** The message broker must be able to support a large scale deployment with minimum infrastructure cost.
- **Timeliness.** The message broker must identify devices according to the latest attributes and device status when requested by the users.
- **Inclusiveness.** The message broker must return a device list that contains every active devices that match the received query.
- **Robustness.** The message broker must be resilient to service failures and high network churn.

In our design, **searchability** allows the message broker to be expressive so that applications and devices can declare their own attribute keys and values. The message broker allows device query to contain tailor-made device searching logic so that the applications have tremendous flexibility to define how to identify the corresponding nodes of interest.

Among the declared attribute keys and values, the applications also have the freedom to declare who are the authoritative source of information for each attribute. Only the authoritative information source is allowed to access and modify certain attribute field of the corresponding devices. Such **verifiability** can effectively prevent malicious attacks such as identity spoofing and eavesdropping.

To achieve **scalability**, the message broker offloads the device status upload workload to the end devices. Some selected end devices will receive status update from others. By maintaining the list of such representative devices, the message broker service is capable of pulling the up-to-date device status when needed.

The message broker can effectively offset the extra workload of frequently updating the dynamic attribute by limiting the scope of message exchanges. This mechanism allows the message broker to offer multi-granular attribute update channels for applications with different **timeliness** requirements. Static attributes such as device affiliation can be updated through a global channel while a dynamic attribute such as geo-location will be exchanged in a smaller scope. In the extreme cases where real-time varying attributes will be collected, devices are no longer exchanging/updating their attributes and status information with other devices. A communication channel will be established among the nodes of interests for pulling the status and attributes directly in an on-demand manner.

In our design, the **inclusiveness** and **robustness** are achieved by strong semantics that regulate and manage the message exchanges among devices. The inclusiveness guarantees that the applications can reach every active node of interest through the message broker. It givens the applications complete view of available services and resources in the globe. Each of the components is designed and implemented as a distributed system which tolerates certain level of failures. More importantly, they are designed to be self-sustainable so that they do not depend on each other to function properly.

Figure 1 shows the architecture of our proposed message broker system that we refer to as EF-broker. EF-broker mainly provides three services: (1) device discovery and inventory management (DDIM), (2) dynamic group management (DGM), and (3) communication channel orchestration engine (CCOE). The DDIM is implemented as a centrally managed, geo-distributed bookkeeping service that maintain a global view of all active devices. It serves as the rendezvous point of newly arrived devices while it also maintains the availability of devices by requiring them to update their attributes and status in a low frequency as heartbeat signals. Such global attribute update channel disseminates the status update messages to geo-distributed DDIM servers in an eventual consistent manner. DDIM is capable of answering device queries that depend on relatively static attribute values efficiently. To keep track of dynamic device attributes whose values change frequently, for each dynamic attribute key, a peer-to-peer cluster of devices that we refer to as dynamic group (DG) are created. Devices in the same DG exchange attributes and status information in a much higher frequency using gossip protocols. Because of the gossip protocols, every device maintains fresh attributes and status of other members in the same group. Representative devices are selected as the entry points of each DG. The representative nodes disseminate and maintain the membership list of the DG in a strong consistent manner in order to achieve inclusiveness. The centrally managed, geo-distributed dynamic group management (DGM) service is introduced to manage the life-cycles of a large number of DGs. It is responsible of creating, terminating, splitting, merging of DGs, as well as maintaining and repairing entry points for DGs. The DGM service provides a more fine-granular attribute update channel by forwarding device queries to the entry points of the appropriate DGs. At last, EF-broker is also capable of creating pub-sub channels among devices in the DGs in an on-demand manner so that applications can pull real-time attributes and status directly from the nodes of interests. The CCOE is introduced to manage the life-cycles of the pub-sub channels.

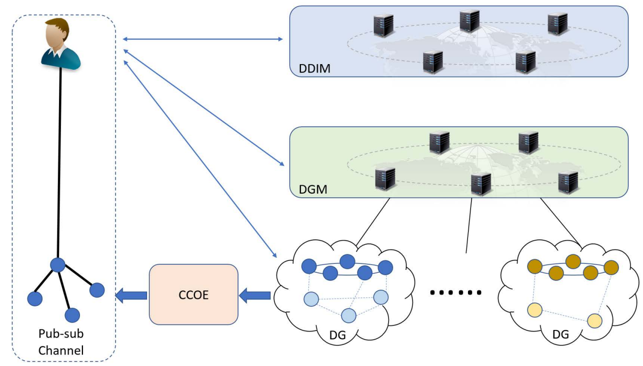

Figure 1: Architecture of EF-broker.

REFERENCES

[1] William Adjie-Winoto, Elliot Schwartz, Hari Balakrishnan, and Jeremy Lilley. 1999. The design and implementation of an intentional naming system. *ACM SIGOPS Operating Systems Review* 33, 5 (1999), 186–201.
[2] Robert J Hall. 2012. A geocast-based algorithm for a field common operating picture. In *MILITARY COMMUNICATIONS CONFERENCE, 2012-MILCOM 2012.* IEEE, 1–6.
[3] Butler W Lampson. 1986. Designing a global name service. In *Proceedings of the fifth annual ACM symposium on Principles of distributed computing.* ACM, 1–10.
[4] Petar Maymounkov and David Mazieres. 2002. Kademlia: A peer-to-peer information system based on the xor metric. In *International Workshop on Peer-to-Peer Systems.* Springer, 53–65.
[5] Abhigyan Sharma, Xiaozheng Tie, Hardeep Uppal, Arun Venkataramani, David Westbrook, and Aditya Yadav. 2015. A global name service for a highly mobile internetwork. *ACM SIGCOMM Computer Communication Review* 44, 4 (2015), 247–258.