Replicating a Rendezvous Node for a Core-based Tree Multicast Protocol in NDN Networks for Providing Low Latency

Yoshiki Kurihara\textsuperscript{1a)}, Yuki Koizumi\textsuperscript{1}, and Toru Hasegawa\textsuperscript{1}

\textsuperscript{1} Graduate School of Information Science and Technology, Osaka University
Yamada-oka, Suita-shi, Osaka 565-0871
\textsuperscript{a)} y-kurihara@ist.osaka-u.ac.jp

Abstract: Since data obtained by Internet of Things devices is correlated with their locations, location-based services of exploiting such locality are promising. An example service is that a driver, i.e., a collector of data, asks cars on the road 2 km ahead of her or him how many cars are driving there. Since such a service requires low latency multicasting communication, this paper designs a low latency publish/subscribe mechanism by leveraging name-based communication like Named Data Networking.

Keywords: Publish/Subscribe communication, Named Data Networking, Multicast Protocol and Low Latency Communication

Classification: Network

References

[1] G. Gartner and H. Huang, Progress in Location-Based Services 2016, Springer, 2016.
[2] “MQTT 3.1 Protocol Specification,” http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html.
[3] J. Chen, M. Arumaithurai, L. Jiao, X. Fu, and K. K. Ramakrishnan, “An Efficient Content Oriented Publish/Subscribe System,” in Proceedings of IEEE/ACM ANCS, pp. 99–110, Oct. 2011.
[4] D. Kim, D. Meyer, H. Kilmer, and D. Farinacci, “Anycast Rendezvous Point (RP) mechanism using PIM and MSDP,” RFC 3446, Jan. 2003.
[5] K. Ryu, Y. Koizumi, T. Hasegawa, “Name-based Geographical Routing/Forwarding Support for Location-based Services,” in Proceedings of IEEE HotPNS 2016, Nov. 2016.

1 Introduction

Data obtained by Internet of Things (IoT) devices is correlated with their locations, and hence location-based services (LBSs) of exploiting such locality are promising [1]. Key requirements to LBSs are summarized as data retrieval from multiple IoT devices and low latency communication. Hereafter,
such data is called location data. Since location data retrieval has inherently a multicasting nature, we choose Content Oriented Publish/Subscribe Communication System (COPSS) [3] which provides publish/subscribe communication in Named Data Networking (NDN) networks for data retrieval. Its name-based communication avoids round trip delays for resolving names of IoT devices to physical addresses.

Despite the advantage of avoiding name resolution, a core-based multicast protocol adopted by COPSS incurs additional hops to the shortest paths between a collector and IoT devices because multicast trees are constructed from a common root called a rendezvous node (RN). The paper reduces such additional hops by placing multiple RNs so that a collector of data and each IoT device rendezvous to their closest RN.

2 Related Work
Publish/subscribe communication protocols are used for IoT data retrieval. MQTT [2] is a protocol in IP networks and a central server plays a role of a RN. However, TCP/IP incurs long latency due to its rich flow control. On the contrary, COPSS [3], which does not have such control and name resolution, provides shorter latency than MQTT. The paper proposes placing multiple RNs to reduce hops inspired by an anycast mechanism in IP networks [4].

3 Architecture
Figure 1 shows the architecture for location data retrieval.

3.1 Naming Scheme
A name of location data is specified by a sequence of a location name and a data name, delimited by the reserved word #dat. A location name is specified according to Z-order as our previous work proposes [5]. A Z-order number is a quaternary number, $z_1z_2\ldots z_m$. The most significant digit $z_1$ corresponds to one of the four squares divided from the largest square. Each square is divided into four squares and they are numbered from $z_10$ to $z_13$. Smaller squares are recursively numbered in the same way. Z-order quaternary numbers are naturally mapped onto name components like /$z_1$//$z_2$/\ldots/$z_m$. An example location data name is /3/0/0/#dat/car/speed.

Fig. 1. Architecture for Location Data Retrieval
3.2 Data Retrieval by Leveraging COPSS

COPSS [3] consists of the two procedures: First, subscribers send a subscribe packet to the RN so that a multicast tree is constructed from the RN to them. Its destination is a location data name, e.g., /3/0/0/#dat/car/speed in Fig. 1. The name of RN, i.e., /RN, is added to it so that the subscribe packet is delivered to the RN. COPSS routers on the path from each subscriber to the RN records a pair of a location data name and an outgoing face in its subscription table (ST). Second, a publisher sends a publish packet to the RN with a location data name as a destination. After receiving it, the RN forwards this packet after removing the RN name. The publish packet is forwarded by downstream COPSS routers according to their STs.

A collector retrieves location data in the two steps assuming that all IoT devices subscribe to their location data names and a collector subscribes to its name, e.g., /Collector. First, a collector sends a publish packet to all IoT devices of which location data names are of interest. Second, each of the IoT devices sends a publish packet with location data to the name.

4 Anycast Mechanism

In the case of native COPSS, a single RN is deployed, as shown in Fig. 1. To avoid additional hops incurred by the single RN, we place a group of RNs at edge routers, as illustrated in Fig. 2, wherein they are called an anycast group.

The idea behind the RN replication is summarized below: A multicast tree which has multiple roots is constructed. Any RN anycasts a received publish packet to the other RNs so that the publish packet is delivered to all the subscribers whichever RN receives it. An important assumption is that a collector retrieves location data which is geographically adjacent to it. Since a geographical topology and a router topology are similar, many pairs of collectors and IoT devices rendezvous to their closest RN to each other.

1) Multicast tree construction: Two types of STs are used for subscribers and other RNs in the anycast group as shown in Fig. 2. This paragraph explains how the two STs of RN1 are updated when it receives subscribe packets. Each RN has a name in the form of /RN/RN1, where /RN is the common prefix of all the RN names. First, a subscriber sends a subscribe packet to /RN, precisely speaking /RN/3/0/0/#dat/car/speed. The packet is forwarded to the closest RN, i.e., /RN/RN1. RN1 records the pair of the location data name and the face #0 at the ST for /RN/RN1 and then anycasts the subscribe packet to the other RNs. Second, when RN1 receives a subscribe packet with the name /RN/3/0/0/#dat/car/speed from other RN, e.g., RN2, it records the pair of the name and the RN’s name, i.e., /RN/RN2, at the ST for /RN.

Although the replication reduces the average path length, it increases the number of ST updates of each RN because subscribe packets triggered by device movements are delivered from the closest RN to the device to all the other RNs. In Section 5, we evaluate their tradeoff relations.
2) Publish packet forwarding: When a collector, e.g., the publisher in Fig. 2, sends a publish packet to a location data name, i.e., /RN/3/0/0/#dat/car/speed, it is forwarded to the closest RN, i.e., /RN/RN1. Then the RN forwards the packet to the faces corresponding to the location data name, and anycasts it to the faces to the other RNs. If the upstream RNs of the publisher and the subscriber are the same, the publish packet is forwarded via the RN which is the closest to both of them.

5 Evaluation through Simulations

This section evaluates path lengths and ST updates through simulations.

5.1 Simulation Conditions

The target area is the special wards of Tokyo, i.e., a square area of 32 km width. The area is recursively divided eight times. The smallest square has an 8-digit Z-order number and its width is 0.125 km. We construct a router-level topology on the basis of the locations and the coverage areas of telephone exchange buildings of NTT East Corporation. The router-level topology is a tree, and its depth is three. The root router, i.e., the first-level router, is placed at the building in Otemachi. The six second-level routers are placed at buildings near the 6 big terminal stations. The third-level routers are placed so that the target area is covered uniformly, and the routers are connected with the closest second-level routers. The fourth-level routers are placed at all the buildings and they are connected with the closest third-level routers.

Moving and stationary devices are deployed as follows: 2.4 million cars, which move at the average speed of 30 km/h, are deployed on realistic roads in Tokyo. Besides, 1 million stationary sensors are randomly deployed. In the evaluation, drivers obtain location data of cars in a circle of 200 m radius located at 2 km ahead.

5.2 Path Stretch Due to RN Deployment

The average path lengths are 3.35, 2.98 and 3.30 in the cases where RNs are deployed at the 2nd, 3rd and 4th-level routers, respectively. The path lengths are averaged over these 1,000 requests and the path length is defined...
as the number of hops from a car to cars with speed sensors. 1,000 cars are randomly selected as collectors. The number of hops to retrieve location data is much smaller than that of the case of the single RN. In the case where a single RN is deployed at the first level router, the number of hops between cars and the RN is four and thus the path length is always eight.

5.3 Frequency of ST Updates

Fig. 3 shows the average frequencies of ST updates which add and delete entries of location data names in each RN in the case of various numbers of cars. The average frequency decreases as the cars’ number increases. It means that the larger the density of cars becomes, the smaller the frequency of ST updates becomes. This is because, in the case that cars are densely deployed, even if all cars move, location data names recorded at edge routers do not frequently change. This reduces the number of subscribe packets forwarded by edge routers to RNs. In addition, even if cars are sparsely deployed, the number of ST updates in a minute is at most 150 in total. Hence, packets generated to update STs is not a serious issue.

![Graph showing frequency of ST updates](image)

Fig. 3. The frequency of ST updates

6 Conclusion

The paper designs an anycast mechanism to enhance a name-based public/subscribe protocol to achieve low latency multicasting.

Acknowledgments

The research leading to these results has partially received funding from NICT under Contract 191.