A Study on Routing-based Mobile Architecture for IoT devices in Cellular Networks

Submitted to
Graduate School of Information Science and Technology
Osaka University

January 2018

Masanori ISHINO

List of Publications

Journal Papers

1. Masanori Ishino, Yuki Koizumi, and Toru Hasegawa, "A Routing-Based Mobility Management Scheme for IoT Devices in Wireless Mobile Networks," *IEICE Transactions on Communications*, vol. E98-B, no. 12, pp. 2376–2381, Dec. 2015.

2. Masanori Ishino, Yuki Koizumi, and Toru Hasegawa, "Relay Mobile Device Discovery with Proximity Services for User-Provided IoT Networks," *IEICE Transactions on Communications*, vol. E100-B, no. 11, pp. 2038–2048, Nov. 2017.

Refereed Conference Papers

1. Masanori Ishino, Yuki Koizumi, and Toru Hasegawa, "A Study on a Routing-Based Mobility Management Architecture for IoT Devices," in *Proceedings of IEEE International Conference on Network Protocols, PhD Forum*, pp. 498–500, Oct. 2014.

2. Masanori Ishino, Yuki Koizumi, and Toru Hasegawa, "Leveraging Proximity Services for Relay Device Discovery in User-provided IoT Networks," in *Proceedings of the 2nd IEEE World Forum on Internet of Things (WF-IoT 2015)*, pp. 553–558, Dec. 2015.

Non-Refereed Technical Papers

1. Yo Nishiyama, Masanori Ishino, Yuki Koizumi, Toru Hasegawa, Kohei Sugiyama, and Atsushi Tagami, "Design of Routing-based Packet Forwarding over an LTE Core Network," in *IEICE Technical Report*, vol. 115, no. 140, IN2015-37, pp. 85–90, Jul. 2016 (in Japanese).

2. Masanori Ishino, Yuki Koizumi, and Toru Hasegawa, “A Study on Leveraging Proximity Services for Relay Device Discovery in User-provided IoT Networks,” in *IEICE Technical Report*, vol. 115, no. 310, IN2015-64, pp. 17–22, Nov. 2015 (in Japanese).

3. Yo Nishiyama, Masanori Ishino, Yuki Koizumi, Toru Hasegawa, Kohei Sugiyama, and Atsushi Tagami, “A Proposal on Routing-Based Distributed Mobility Management for 5G ,” in *IEICE Technical Report*, vol. 116, no. 231, IN2016-44, pp. 47–52, Sept. 2016 (in Japanese).

Preface

Internet of Things (IoT) applications such as environmental monitoring and big data analysis based on sensor data obtained from IoT devices have been becoming popular. Although some of such IoT devices are mobile devices like smartphones, their mobility communication patterns are different from those of traditional mobile devices. They have different communication patterns from the traditional ones including low mobility, intermittent data transmissions, infrequent transmission, and so on. Another important feature is that all IoT devices are not always directly connected to the Internet. It means that in an IoT era, the mobile architecture should support mobile patterns of two types of IoT devices: *cellular IoT devices* like smartphones with sensors and *non-cellular IoT devices* without cellular interfaces like tiny and resource-constrained IoT devices. Introduction of IoT gateways which accomodate non-cellular IoT devices to a cellular network is inevitable.

The special communication patterns come from new types of application which are different from one-to-one real-time communications. Typical applications in the IoT ear are urban sensing applications for environment monitoring, which require deployment of large number of IoT devices some of which are stationary devices as data sources in wider metropolitan areas. Since IoT devices have the above characteristics such as low mobility and intermittent transmissions, current most mobile network architectures supporting mobile devices such as cellular phones and vehicles are over-engineered for mobility management of such IoT devices.

New applications in an IoT era require more light-weight mobility management than strict mobility management in a smartphone/cellar phone era, wherein session continuity is an inevitable requirement. This thesis reconsiders mobility management for IoT devices without focusing on session continuity. The author designs the mobilr architecture which is based on more light-weight mobility management scheme than that in a cellular network according to the following approaches: First, the thesis adopts a routing-based mobility management scheme for managing mobility of celular IoT devices. An important research challenge is to how to design compact data structures of routing information because movements of a huge number of IoT devices and IoT gateways are propagated trigger such routing information updates. The routing-based approach is inspired by the fact that such IoT devices are not likely to move as often as ordinary smartphones. Second, the thesis designs a discovery mechanism for an IoT gateway which is connected to a target non-cellular IoT device.The architecture introduces an IoT gateway, called a relay user equipment (UE) device, which connects non-cellular IoT devices to a cellular network. Its responsivity is managing mappings between non-cellular IoT devices and IoT gateways because such mappings are not permanent. Since non-cellular IoT devices cannot communicate with a base station, how they communicate each other and how to discover an IoT gateway which connects to a target IoT device are an important research issue. The micro mobility management provides such discovery by leveraging a LTE-based device-to-device (D2D) communication mechanism.

The contributions of the thesis are summarized below:

1. The author re-designs mobility management suitable to IoT devices, whereas most IoT communication protocols use existing mobility management schemes provided by cellular and mobile IP networks.

2. The author proposes a new routing-based mobility management scheme, which can achieve low overheads in terms of size of routing tables and route update messages, for low-mobility IoT devices.

3. To accommodate non-cellular IoT devices in cellular networks without wasteful resource consumptions of UE devices and base stations, the author designs and evaluates an IoT gateway/relay UE discovery mechanism by leveraging a LTE-based D2D communication mechanism.

Acknowledgments

First of all, I would like to express my greatest appreciation to my supervisor, Professor Toru Hasegawa, Graduate School of Information Science and Technology, Osaka University, for his insightful comments and valuable advice. He carefully reviewed my papers and gave me much advice for completing this thesis. Without his proper guidance and persistent help this thesis would not have been possible.

I would like to express my gratitude to the members of my thesis committee, Professor Masayuki Murata, Professor Takashi Watanabe, and Professor Teruo Higashino, Graduate School of Information Science and Technology, Osaka University, and Professor Morito Matsuoka, Cyber Media Center, Osaka University, for carefully reviewing this thesis and providing valuable comments.

I am deeply grateful to Professor Hiroyuki Ohsaki, Department of Informatics, School of Science and Technology, Kwansei Gakuin University, and Assistant Professor Sho Tsugawa, Faculty of Engineering, Information and Systems, University of Tsukuba, for irreplaceable guidance and support. Especially, I would like to express my deepest appreciation to Professor Hiroyuki Ohsaki. His excellent guidance on research motivated me to become a researcher.

I am cordially thankful to a member of research meetings, Assistant Professor Yuki Koizumi, Graduate School of Information Science and Technology, Osaka University for valuable comments to raise the quality of my research.

I also express my appreciation to all members of the Information Sharing Platform Laboratory, Graduate School of Information Science and Technology, Osaka university. I am thankful to Yo Nishiyama for his important contribution. I also am thankful to Ms. Tomoko Ueshima for her kind help.

Finally, I would like to express my gratitude to my loving family members and relatives for their heartily and continued support. In particularly, I want to thank my parents, Masahiro Ishino and Keiko Ishino, for their warm support and encouragement.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

In an IoT era [1–3], huge number of IoT devices whose mobility and communication patterns are different from traditional mobile devices such as laptop computers and cellular phones are deployed for various types of IoT applications. They have different communication patterns from the above traditional ones including low mobility, intermittent data transmissions, infrequent transmission, and so on [4]. Another important feature is that all IoT devices are not always directly connected to the Internet [5]. The special communication patterns come from new types of application which are different from one-to-one real-time communications. Typical applications in the IoT ear are urban sensing applications for environment monitoring [6], which require deployment of large number of IoT devices some of which are stationary devices as data sources in wider metropolitan areas. Since IoT devices have the above characteristics such as low mobility and intermittent transmissions, current most mobile network architectures supporting mobile devices such as cellular phones and vehicles are over-engineered for mobility management of such IoT devices.

New applications in an IoT era require more light-weight mobility management than strict mobility management in a smartphone/cellar phone era, wherein session continuity is an inevitable requirement. This thesis reconsiders mobility management for IoT devices without focusing on session continuity.

1.2 Approaches

Designing a new mobility management scheme which is suitable for IoT applications is an important research challenge. In this thesis, the author addresses two problems specific to IoT communications: mobility management overheads for a large number of IoT devices and resource-constrained IoT devices which do not have direct access to the Internet. The author addresses the two problems assuming that cellular networks are used to accommodate IoT devices both which have cellular interfaces and which do not have them. The first issue is how to manage locations of IoT devices with cellular interfaces which do not move as frequently as smartphones. We call such IoT devices just IoT devices, hereafter, We call such mobility management *macro mobility management* because it is performed in an cellular network scale. The second issue is how to manage IoT devices without cellular interfaces, which we call *non-cellular IoT devices.* Since non-cellular IoT devices are mostly stationary, how to discover a gateway which connects a non-cellular IoT device to a cellular network is an important research issue. We call such management *IoT gare way discovery/relay user equipment (UE) discovery*. Hereafter, the author calls such a gateway either an IoT gatrway or a relay (UE) device and use them interchangeably.

1.2.1 Macro Mobility Management

The first issue in the thesis is macro mobility management which leverages low mobility of IoT devices. The author revisits current mobility management schemes based on the identifier/locator separation concept. To address devices' mobility strictly, the current mobile architecture uses strict mobility management [7, 8]. The word strict means that the architecture keeps mappings between devices' identifiers and their locations and update the mappings whenever the devices change their attached locations to networks. If current mobile networks accommodated a huge number of IoT devices in the IoT era, it would result in the large size of routing information held as the mappings in network devices and quite frequent communication for updating route information. The fact that devices' identifiers are not aggregated as network addresses like IP addresses are aggregated incurs large communication overheads for mobility management.

To avoid the above overheads, the author proposes a new routing-based mobility management scheme for IoT devices, which breaks with the old idea of identifier/locator separation and aggressively aggregates device's identifiers [9, 10]. This is because a huge number of the identifiers which are not aggregated is a main obstacle to reduce the size of exchanging routing information. The author leverages a Bloom Filter as a data structure to store routing information at network devices inspired by the fact that studies on efficient longest prefix and name matching in IP and NDN networks [11–14], respectively.

1.2.2 IoT Gqteway Disvovery

The second issue is how to connect resource-constrained non-cellular IoT devices to the Internet. In this thesis, assuming that cellular networks are everywhere and all IoT devices have local wireless interfaces such as Wi-Fi and Bluetooth, the author proposes to introduce a new user-provided network concept into cellular-based IoT networks. The idea is inspired by the fact that crowdsourcing applications based on cellular networks have gathered attention because smart phones are regarded as data sources for various types of crowdsourcing urban sensing applications [15]. The author regards a smart phone as an IoT gateway for non-cellular IoT devices which do not have a cellular interface rather than a data source. If non-cellular IoT devices, which have dedicated sensors for environmental or road monitoring and Wi-Fi interfaces, are deployed and smart phones are used to upload sensed data collected by the IoT devices via cellular network, the author believes that these applications would become more attractive. In proposed user-provided IoT networks for sensing applications, the author assumes that sensed data retrieval from IoT devices through IoT gateways is performed on request basis to prevent large wasteful resource consumptions of IoT devices, IoT gateways, and cellular base stations. Under the above assumption, user-provided IoT networks need schemes to discover IoT gateways, which connect with issued IoT devices using Wi-Fi connections.

An important research issue is how to discover an IoT gateway which uploads sensed data since such an IoT devices does not have a routable address provided by a cellular network. The author proposes a light-weight relay UE discovery scheme, i.e., IoT gatway discovery, by leveraging proximity services (ProSe) [16–18] which is one of device-to-device (D2D) communication architectures. One of key features of the ProSe is natively implementation of device discovery functions, which can change its communication range depending on the size of targeted discovery areas and performed by both smart phones and cellular base stations. Due to this feature, both whole-cell and local discovery processes can be performed depending on existence of the most recent relay mobile devices' information to achieve low overheads in terms of discovery messages.

1.3 Routing-Based Mobility Management:Macro Mobility

In Chapter 4, the author designs a routing-based mobility management scheme for macro mobility. The scheme leverages IoT device communication features including low mobility, intermittent transmissions of small data, infrequent transmission, etc.

Most traditional mobility architectures, which strictly manage devices' mobility, embrace a common concept of identifier/locator separation so that the device is identified by its identifier and that packets are routed according to its locator [19]. Even if a mobile device changes its location, it is able to access it by updating a mapping between its identifier and its new locator. In these architectures, the mapping is managed for each device, since the devices' movements are independent of each other. The seamless access pays the serious cost of the fatal defect that mapping cannot be aggregated. This results in the large size of routing information.

Chapter 4 revisits current mobility management schemes for wireless mobile networks based on identifier/locator separation. The author breaks with the separation, and aggressively aggregate identifiers themselves leveraging IoT communication patterns. For example, since IoT devices deployed densely to sense a target area upload similar data packets repeatedly, all of them need not to be delivered. This loose requirement enables routers to advertise imprecise routing information.

The author proposes a new routing-based mobility management scheme for IoT devices. As the first step to design the scheme, the author focuses on wireless mobile networks which are well managed by mobile network operators. The author addresses the following research question: how to reduce the size of routing information associated with IoT devices. The research question is raised because a huge number of identifiers assigned independently of devices' locations is a main obstacle to reduce the routing information size. The author adopts a Bloom Filter [11] as a data structure to store routing information at routers inspired by the fact that studies on efficient IPv4 and IPv6 longest prefix matching [20] do so. The Bloom Filter enables to reduce the routing information size at the cost of its impreciseness, i.e., their false positive probability.

The contributions of this thesis are summarized as follows: (1) This thesis is one of the first studies of adopting the Bloom Filter to a mobility management scheme to reduce the routing information size as far as the author knows. (2) The proposed routing-based mobility management scheme eliminates strict mobility management by shedding light of IoT communication patterns. The fact that their communications do not need accuracy in, i.e., a high delivery rate, enables to do so. (3) Feasibility of the proposed scheme is validated through simulations for a regional-scale wireless mobile network. (4) Finally, this thesis designs a routing protocol based on the proposal by extending Open Shortest Path First (OSPF) so that the routing-based mobility management provides scalality required by cellular networks. The simulation results show that the proposed scheme of replacing current mobility architecture with a routing-based one has scalability for cellular networks.

1.4 Relay UE/IoT Gateway Discovery:Micro Mobility

Crowdsensing applications based on cellular networks [21–23] have gathered attention because smart phones having cameras and sensors, e.g., accelerometers, lights, compasses, temperature instruments, are seen as data sources for various types of crowdsensing urban sensing applications. Urban sensing applications [15, 24, 25] are promising in metropolitan areas because the number of smart phones is so large that sensed data is collected from all over the areas. If smart phones were used to upload sensed data collected from fixed Internet of Things (IoT) [1, 2] devices dedicated to environmental or road monitoring in terms of variety of applications and granularity of locations of sensed data, such crowdsensing applications would become more attractive. The author proposes user-provided IoT networks for the above crowdsensing applications, wherein smart phones provide connectivity to cellular networks for fixed IoT devices which have only a local wireless interface, e.g., Wi-Fi. A smart phone called a user equipment (UE) device, hereinafter reffered to as *relay UE devices*, forwards packets to and from the IoT device.

User-provided IoT networks in urban areas have a potential to retrieve on demand a fresh and fragile data piece from an IoT device of interest, whereas most crowdsourcing applications provide application users with instructions obtained by analyzing collected data and thus smart phones and IoT devices upload sensed data at their own initiatives. In other words, a server or an application user does not request them to upload sensed data periodically. This quick retrieval enables new type of applications such that emergent information about accidents and traffic congestion, which would not be provided by conventional crowdsourcing applications. On the contrary, this chapter focuses on mechanisms for directly requesting such non-cellular IoT devices.

A key technique is to discover relay UE devices which connect to non-cellular IoT devices of interests because the IoT devices are not managed directly by cellular networks. Such discoveries are not trivial due to the following reasons: First, cellular communication does not provide discovery functions which can be used for discovering relay UE devices without wasteful resource consumptions. Although LTE downlink multicast services, e.g., multicast-broadcast single-frequency network (MBSFN), are candidate solutions, they can not set their discovery ranges to the smaller area than a cell. Such broadcasting wastefully consumes radio resources and UE devices' batteries in the case that all UE devices need not be searched because a candidate relay UE device exists in the near distance from an IoT device. Second, the short range of Wi-Fi makes the discovery difficult even in the case that a base station of a cell knows that a relay UE device exists in the cell a short while ago. Since the possibility that a new relay UE device is within the range of such an old relay

UE device is low, the old relay UE device does not discover the new relay UE device within its Wi-Fi range. In this case, a base should do discovery of the new relay UE device in the whole cell.

In this thesis, the author designes a relay UE discovery mechanism for the mobile architecture based on named data networking (NDN) [26] and the evolved packet system (EPS), which consists of the evolved packet core (EPC) and evolved universal terrestrial radio access network (E-UTRAN), of long term evolution (LTE) [27].

A key mechanism of the architecture is a light-weight relay UE discovery mechanism by leveraging proximity services (ProSe) [16, 18, 28], which is one of device-to-device (D2D) communication architectures. It adopts a *ProSe Direct Discovery (Model B)* function of the ProSe [18] based on D2D communications for relay UE discoveries because this function provides a broadcasting function of which range is far larger than that of Wi-Fi and its broadcasted messages can contain additional information such as identifiers of target IoT devices. A key feature of the proposed mechanism is to reduce the number of messages for relay UE discovery by leveraging the longer ProSe communication range than that of Wi-Fi. This intermediate range enable our discovery mechanism to achieve low overheads in terms of discovery messages.

This thesis generalizes the proposed mobile architecture with the UE discovery mechanisms in terms of name-based communications and target applications, whereas the previous work [29] focuses on the relay UE discovery mechanism itself. The thesie designs and evaluates the discovery mechanism more rigorously. New contributions of this thesis are summarized as follows:

1. To clearly show how existing crowdsensing-based applications are improved, the author focuses on use scenarios wherein non-cellular IoT devices are used as external sensors to collect data sensed by IoT devices other than UE devices.

2. This chapter designs and evaluates a relay UE discovery algorithm more rigorously, which ensures effectiveness and maturity of the proposed algorithm.

1.5 Outline

In this thesis, the author proposes the new mobile architecture with a scalable routing-based mobility management mechanism and a relay UE/IoT gateway discovery mechanisms for IoT applications, which require huge number of fixed and low-mobility IoT devices. In addition, the thesis evaluates the mobility management and relay device discovery mechanisms rigorously to ensure their effectiveness and maturity.

The rest of the thesis is organized as follows. Chapter 2 summaries the realted work. Chapter 3 explains a framework for mobility management for IoT applicatiosns. Chapter 4 explains a routing-based mobility management mechanism. Chapter 5 explains a relay mobile device discovery mechanism with proximity services for IoT devices. Finally, Chapter 6 concludes the thesis.

Chapter 2

Related Work

The thesis is compared with the two series of studies: mobility managemant and device discovery.

2.1 Mobility Management

2.1.1 Mobility Architectures for Mobile Networks

Many mobility architectures for wireless mobile networks have been proposed, and they use the common concept that they separate identifiers and locators of devices [19]. The concept requires mapping between identifiers and current locators, and a couple of approaches are proposed in the literature: *using anchor nodes* and *using global name-to-address (location) resolution services* [30].

The approaches using anchor nodes always track devices' physical locations, and maintain and update mappings between identifiers and locators of the mobile devices to provide tunnels to them. For example, in the Mobile IP [7, 8] architecture, a router called "Home Agent" maintains a mapping between a home address as an identifier and a care-of address as a locator for each mobile device. As another example, in an long term evolution (LTE) network, a home subscriber server maintains a fixed international mobile subscriber identity as an identifier and a list of tracking areas (TAs) as a locator to manage tunnels between a packet data network gateway and the mobile device [27].

The approaches using global name-to-address resolution services maintain and update mappings between identifiers, whose namespace is flat, to locators of *all* mobile devices. Devices and routers can use the name-to-address resolution service to resolve names of destination devices to the addresses depending on their locations. For example, MobilityFirst architecture [31] provides global name resolution service to resolve device names to network addresses of networks which the devices belong to.

These two approaches suffer from large memory for maintaining *all* mobile devices' states and communication overheads for updating mappings every time when the devices move. For instance, when a Mobile IP network accommodates N mobile devices, its home agent has to store N mappings and handle N devices' movement using control messages, that is, when the N devices move, the home agent has to handle N messages to update their mappings. In contrast, our scheme can eliminate the need for the large memory for maintaining *all* devices' states by aggregating them using Bloom Filters. It can also reduce the number of control messages to update the states by periodically exchanging the aggregated states as the Bloom Filters.

2.1.2 Distributed Mobility Management

Many studies have addressed a scalability problem of centralized mobility management, i.e., traffic concentration at a mobility anchor in a core of a cellular network. Distributed mobility management is a promising solution to avoid the traffic concentration by moving mobility anchor functions such as packet forwarding and location management to multiple edge routers. Two types of distributed mobility management mechanisms are studied in the literature. First, a name resolution scheme [32, 33] moves a packet forwarding function to edge routers, whereas it keeps a location management function in a dedicated entity in a central office. The location management is performed by resolving an identifier of an MN to its current locator and this name resolution incurs round trip latency before sending a data packet to an MN.

Second, an anchor-less scheme [34–37] moves the both functions of mobility anchor functions to edge routers, and each MN uses one of the edge routers as an *anchored edge router*, which tracks the MN's current location and forwards all data packets destined to the MN. However, an MN receives data packets from its anchored edge router using a tunnel between the anchored edge router and a current edge router and thus the anchor-less scheme incurs redundant traffic at backhaul links. Since bandwidth-demanding mobile applications such as videos are disseminated in 5G era, the traffic concentration at backhaul links is a critical issue.

This thesis proposes a routing-based scheme [10] as the third type. The routin-based scheme moves all mobility anchor functions to edge routers and makes MN movement be advertised by a routing protocol. Although this scheme is fully distributed so that path stretch and traffic concentration are fully avoided, it incurs a large routing overhead. A Bloom Filter is proposed to be used as a FIB to reduce the size of FIBs [14] and then for mobility management [10].

2.1.3 Applications of Bloom Filters in Networking

A Bloom Filter has been widely used in the research area of networking [11, 13]. Tarkoma *et al.* [13] summarized the basic Bloom Filter and its variants, and then classified applications of the Bloom Filters into some categories (e.g., caching, peer-to-peer (P2P), packet routing and forwarding, and security applications). This paper is classified as a packet routing and forwarding application.

Among them, Yu *et al.* [12] propose a Bloom filter-based forwarding architecture for large-scale enterprise and data center networks. They mainly focus on the large size of forwarding information base (FIB) in a router and reduce its size by using a Bloom Filter. In contrast, this paper focuses on reducing the size of route update information exchanged among routers as well as that of a FIB. This paper enables to aggregate Bloom Filters including route update information pieces by the bitwise OR operation of them. This reduces both the sizes of FIBs and route update information.

Blanni *et al.* [38] propose a similar aggregating mechanism for avoiding denial-of-service (DOS) attacks. Each host declares a constraint which specifies a legitimate path to it and Bloom Filters which contain the constraints are created by edge routers. They are aggregated by core routers using the bitwise OR operation like this paper. The basic idea of our and their papers are similar, but our paper applies an aggregating mechanism to a FIB, which is a key data structure for forwarding, whereas their paper applies to an access control list (ACL), which is a supplement one.

In our previous study [9], we proposed a key mechanism of our routing-based mobility management scheme using Bloom Filters. In this paper, we extend our previous study by considering deployment issues. The key difference of this study from the above previous studies is that this paper seamlessly integrates Bloom Filters into a routing protocol as route update information whereas the previous studies only focus on reduction of the size of data structures which have many entries for individual networking applications. Representing the route update information as Boom Filters simultaneously solves the both scalability issues about the size of mapping/route update information and the size/number of control messages for advertising them. In addition, this paper quantitatively evaluates packet delivery rate degradations due to Bloom Filters' false-positives when devices move.

2.2 Discovery

As far as we know, discovery of a mobile IoT gateway for a non-cellular IoT device, i.e., a relay UE device in this paper, is a new research issue. There is no study which has the same goal, i.e., reduction of gateway discovery overheads, of this paper. However, the following two series of studies share a part of the motivation of this paper.

2.2.1 Crowdsourcing-based Urban Sensing Applications

Many types of urban sensing applications have been proposed [15, 24, 25]. These applications mainly use smart phones' embedded sensors as data sources and some of them also use external sensors, which is well-connected to smart phones, such as wearable sensors. In [39], Perera *et al.* have proposed an on-demand mobile distributed sensing platform called C-MOSDEN which allows smart phones to use external sensors as additional data sources. Sensed data is retrieved on request basis, and it is similar to our architecture. In the C-MOSDEN platform, they assume that external sensors are well-connected to smart phones such as sensors for healthcare applications, so that a gateway discovery mechanism is not absolutely necessary unlike our architecture. The most crucial difference in roles of smart phones between previous and our works is that smart phones in our architecture can behave both as mobile IoT gateways.

2.2.2 Smart Phone-based IoT Gateway

A smart phone-based mobile IoT gateway architecture has been proposed by Zachariah *et al.* [5]. Their motivation is to develop an open IoT platform which enables it for everyone to develop various types of IoT applications using IoT devices already deployed by others, whereas most existing IoT platforms are application dependent and thus only the enterprise or the organization can develop proprietary applications. Thus they propose an independent architecture.

The motivation of this paper is similar to their study. The main differences between their and our architectures are as follows: (1) Their architecture mainly focuses on the mechanism to transmit data from IoT devices to cloud servers only when they decide to do so. This means that hosts and cloud servers cannot explicitly request data to IoT devices. This is explicitly raised as one of research questions by their paper. This unsolved research question is precisely a goal of our paper. In other words, as far as we know, there is no study on discovery of a mobile IoT gateway in cellular environments. (2) IoT devices are identified by IPv6 addresses in their architecture whereas they are identified by names. The difference is which naming mechanism is used between name resolution or name-based routing.

Chapter 3

Framework

3.1 Service Model and Target IoT Applications

This section describes assumptions on who provide such target IoT applications and how IoT devices are deployed.

First, this section addresses a service model. An important assumption on IoT services is that some agency or enterprise deploys IoT infrastructure wherein various types of sensors, i.e., IoT devices, are deployed everywhere in urban areas to obtain sensed data for traffic or environment monitoring. Hereafter, the author calls such agencies or enterprises *service providers*. Service providers manage IoT devices and they are generally different operators from cellular network operators. A motivation behind the separation is to make such IoT infrastructure public to all, but service providers do not allow public users to take free ride on cellular network operators' proprietary backbone networks to the Internet. Key features of such separation are described below: First, a service operator provides IoT based services to its customers and buys communication services from cellular network operators. It means that the operator is regarded virtual mobile network operator. Second, a cellular operator provides a service operator with network resources. For example, a network slice is prepared for the service operator.

Second, this section addresses a feasible and inexpensive solution for IoT device deployment. The thesis assumes smartphones as IoT devices because they have many sensors like cameras and acceleration sensors. Off course, sensors paired with smartphones are also IoT devices. However, since it is not rare that all IoT devices have expensive cellular interfaces, the thesis assumes that some of IoT devices only have inexpensive local wireless interfaces, e.g., Wi-Fi interfaces. Thus inexpensive way of accommodating such IoT devices to a cellular network. The thesis leverages

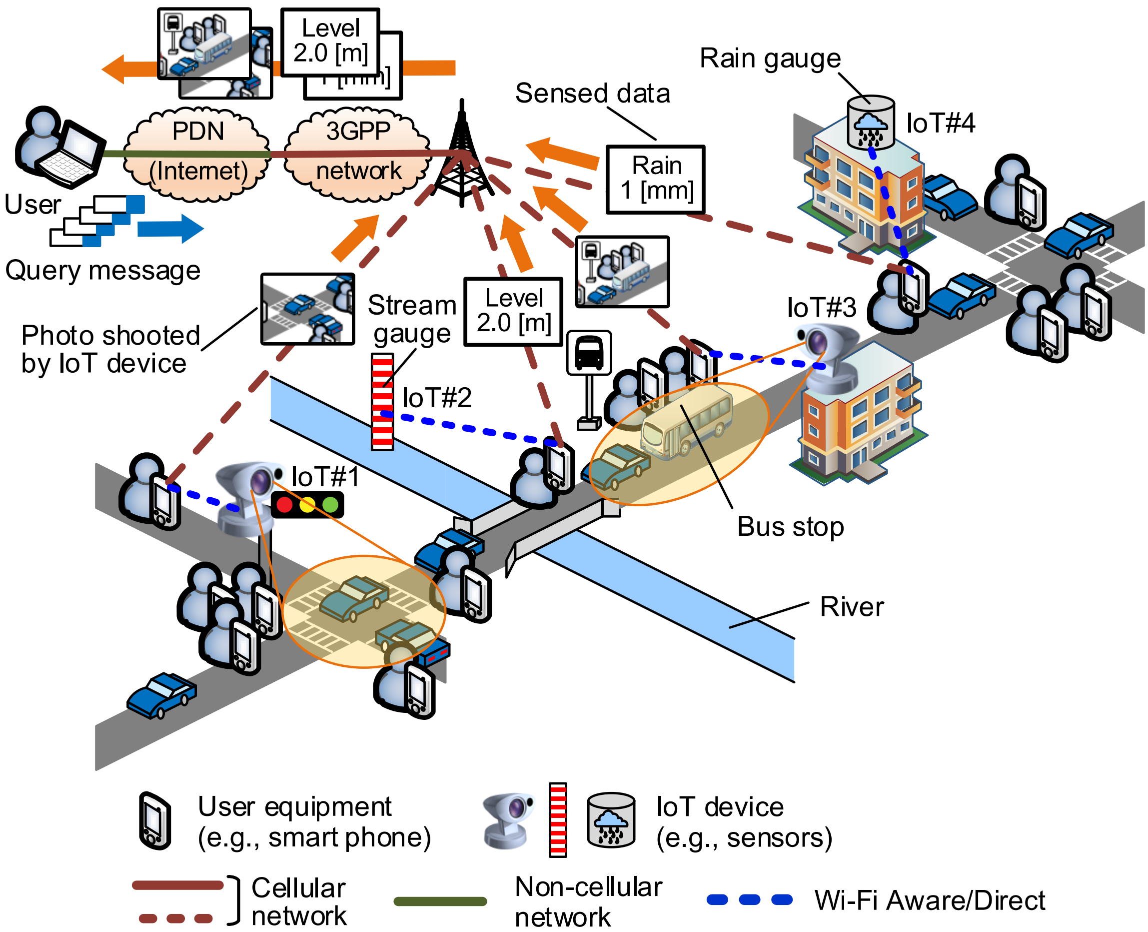


Figure 3.1: Target IoT application. Application users obtain real-time sensed data from non-cellular fixed IoT devices via UE devices of pedestrians.

user-provided network techniques so that volunteer smartphones work as IoT gateways for such IoT devices. Precisely speaking, the thesis assumes some incentive mechanism which encourages such smartphones. The thesis does not address such incentive mechanisms and assumes that the existing work is used.

The thesis does not assume periodical monitoring of data initiated by IoT devices, but real-time retrieval initiated by users. Such realtime and urgent retrieval is important for mission critical IoT applications. An example application works as follows: When an application user needs to obtain sensed data produced by some IoT devices, e.g., sensors or webcams, immediately, he or she sends a request message to the target IoT device. An example of emergent application is that a local agency like a police department traces a suspicious person using video/picture data obtained by users' smartphones and webcams on the street. The request message is forwarded to a relay UE device which works for the target IoT device through packet data networks (PDNs), i.e., the Internet, and cellular networks. Data messages as responses to the request messages are sent back to the user along the inverse path provided by NDN.

Advantages of the service model are as follows:

1. The first advantage is that new network infrastructure need not be deployed. Existing environmental infrastructures are used without new investments to collect sensed data.

2. The second one is traffic reduction based on request based message retrieval. Many types of IoT architectures assume that they intermittently upload sensed data to cloud servers to avoid IoT device discoveries. On the contrary, the proposed discovery mechanism prevents IoT devices from uploading large-size data intermittently.

3.2 System Model and Implementation

After describing the system model, this section designs the mobile architecture to accommodate both mobile and stationary IoT devices.

3.2.1 System Model

The system model is defined as below:

- IoT service providers use Content Centric Networking (CCN)/Named Data Networking (NDN) as the networking architecture and thus all IoT devices, smartphones and users' devices are CCN/NDN capable.

- Each IoT device has a unique hierarchical name according to CCN/NDN. Identifiers assigned independently on router topologies and are not aggregated to prefixes.

- A cellular network is deployed based on IP and thus CCN/NDN packets are capsuled by UDP/IP packets in a cellular network.

- A cellular network provides an IoT service provider with its own slice by leveraging Network Function Virtualization (NFV) techniques as described in Section 3.2.2.

- Two types of IoT devices are used: an IoT device with a cellular interface and an IoT device without a cellular interface. These devices are called IoT devices and non-cellular IoT devices, respectively. Non-cellular devices have local wireless interfaces including Wi-Fi and Pro-SE interfaces. This thesis assumes that IoT device are smartphones with sensor devices. It means that the mobile architecture does not focus on supporting fast movement of vehicles like cars.

- Non-cellular IoT devices are placed at some places and they do not move. The thesis assumes tiny and resource-constrained IoT devices.
- A cellular network is used to accommodate IoT devices. Hosts of users are accommodated by the Internet and obtain sensed data via a cellular network.
- A smartphone works as an IoT gateway which connects non-cellular IoT devices to a cellular network. Mapping between an IoT gateway and non-cellular IoT devices are dynamically determined. We call an IoT gateway a realy UE device, too.
- The Internet is assumed to be based on NDN. An inter-domain routing protocol or a name resolution server forwards a packet to a gateway router of a routing domain of the proposed mobility management architecture.
- Router topologies are arbitrary.

The mobile architecture is deployed over a cellular network and it is responsible for packet forwarding between cellular/non-cellular IoT devices and an edge of a cellular network, i.e., a P-GW. The architecture is assumed to be implemented as an overlay network over a cellular network. After describing how an overlay network is implemented in the next subsection 3.2.2, the author describes the functions of the mobile architecture in Section 3.3.

3.2.2 Overlay Network Implementation

Introducing the proposed mobility management architecture into cellular networks is not trivial because standardization bodies about cellular communications are conservative about using distributed techniques. Thus we take an overlay approach such that a network based on the proposed architecture is overlaid on an Evolved Packet Core (EPC) network of a 3G/4G/5G cellular network. We leverage Network Function Virtualization (NFV) techniques to independently run these two networks as independent slices. An example network configuration is illustrated in Figure 3.2. All routers in a cellular network are NFV-enabled and two slices are run on the routers: A *routing slice* runs the proposed mobility management architecture and an *EPC slice* runs the IP protocol which encapsulates signaling messages and data packets of EPC nodes such as Evolved NodeBs (eNBs), Mobility

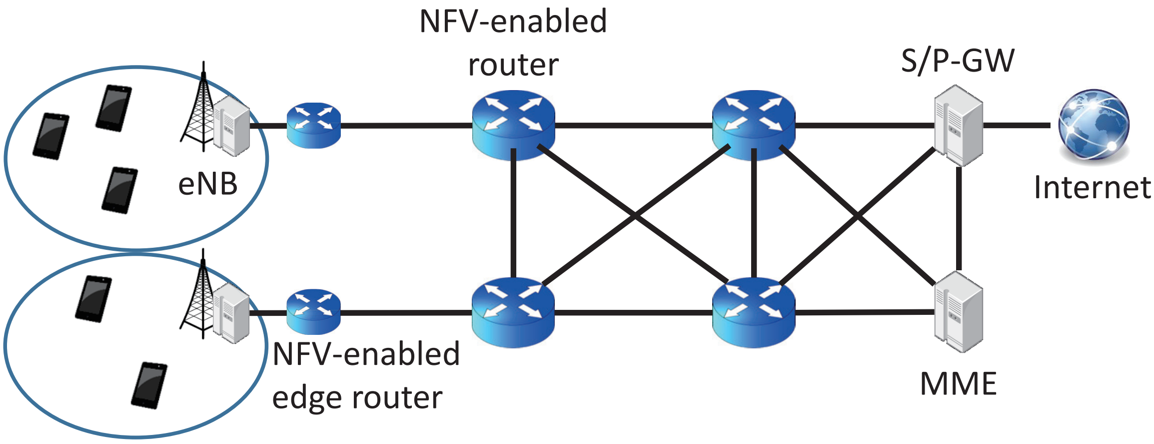


Figure 3.2: An example of network configuration

Management Entities (MMEs) and Packet Data Network Gateway (P-GW). The two slices share EPC nodes and physical/datalink resources of radio networks.

The routing slice implements distributed mobility management and radio bearer management at edge routers instead of an MME, whereas it uses authentication and identifier assignment functions of an MME and a P-GW. That is, an edge router works as a proxy node for an MME and a P-GW or it works as these nodes. As a proxy node for an MME and a P-GW or as these nodes, an edge router sends and receives all signaling messages that would be exchanged between an eNB and an MME or a P-GW. Functions of an edge router are summarized below:

First, an edge router works as a proxy node for an MME and P-GW for attaching an IoT device to an EPC network. When an IoT device is turned on, it sends a signaling message for *attaching* itself to an EPC network. The edge router receives this signaling message and forwards it to an MME and a P-GW. The MME authenticates the IoT device and assigns Global Unique Temporary Identifier (GUTI) to the IoT device and the P-GW assigns an identifier for communications of data packets to the MN. A GUTI uniquely identifies an IoT device in an EPC network and is used to pass signaling messages according to LTE standard procedures.

Second, it works as an MME rather than a proxy node for radio bearer management. It establishes and releases a radio bearer channel between an IoT device and an eNB on behalf of an MME. For example, when an edge router receives a data packet, it sends a signaling message for requesting a radio bearer channel to an eNB.

3.3 Mobile Architecture

As described in Section 2, there are three types of network nodes which manage mapping devices' identifiers/names and current locations, e.g., network addresses like IP addresses: (1) the anchor node [7,8,27], (2) (multiple)edge nodes [9,10,34–37] and (3) distributed name resolutions servers [32, 33]. In addition, there are two types of updating mappings for (2) edge nodes (routers); (2-1) tunnel-based and (2-2) routing-based management. Four types of mobility management schecmes, i.e., schemes of updating mappings betwenn identifiers/names and current locations of IoT devices. The thesis adopts the (2) routing-based management by comparing them from the viewpoints of both users and cellular nework operators as described below. Precisely, they are compared in terms of latency for retrieving data and extra traffic incurred by mobility management. Table 3.1 compares the four approaches assuming that the number of IoT devices is N and each edge router accomodates M IoT devices.

Table 3.1: Copmarison of Mobility Management Approaches

	# of states(anchor/core)	# of states(edge)	Low Latency	Without Extra Traffic
(1)Anchor	O(N)	O(M)		
(2-1)Tunnel,Edge	-	O(M)	Yes	Yes
(2-2)Routing,Edge	O(N)	O(M)		
(3)Name Resoultion	O(N)	-	Yes	

First, the distributed name resolution approach (3) is not appropriate for the target applications. Before a user sends a request packet to an IoT device, she or he should resolve a name of data to an address of an IoT device, which incurs round trip latency. This extra latency is not appropriate for realtime and urgent retrieval of data. Second, the anchor approach (1) is used by current cellular networks [27] is not appropriate for either low latency and extra traffic. The fact that all packets traverse the anchor has bad effects on both latency and traffic concentration. The hop number increase due to the anchor increases latency of retrieving data and apparently traffic is concentrated at the anchor, which is extra traffic for a cellular network. Third, the distributed tunnel-based approach (2-1) alleviates the two problems, but still suffers from hop number increase and extra traffic at each distributed anchor/edge router. On the contrary, since there is no anchor for the routing-based approach (2-2), no extra latency and extra traffic concentration occurs.

The mobile architecture is designed according to the routing-based approach. It is illustrated in Fig. 3.3 and consists of the two mechanisms: *macro mobility management* and *IoT gateway/relay UE device discovery* at edge routers. The macro mobility management mechanism, which is cooperatively

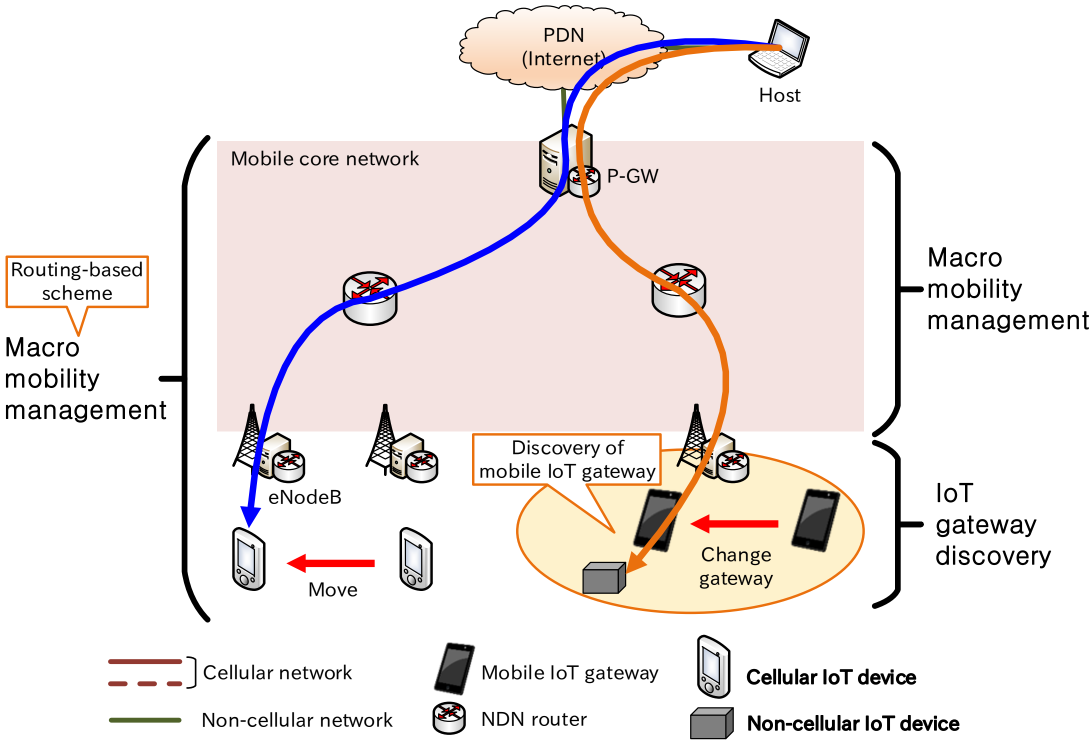


Figure 3.3: An overview of proposed mobility management architecture based on macro mobility management and device discovery

performed by all routers in a cellular network, manages a current location, i.e., an edge router which is conneted to an IoT device, i.e., a smartphone. Since an eNB plays a role of an edge router, it manages mobility among cells. Current locations are propagated to all routers in a cellular network according to a routing protocol. The device discovery mechanism at each edge router records IoT devices which are currently connected to it. Recording an IoT device is trivial since it has a cellular interface. On the contrary, recording non-cellular IoT devices is not trivial because IoT gateways which accomodates the non-cellular IoT devices. Non-cellular IoT devices need be discovered when their IoT gateways are recoreded at the edge router,

The research issues for the both mechanisms are summarized below: First, as shown in Table 3.1, the number of all routers is proportional to that of cellular devices. Since the number of IoT devices in the IoT era would be much larger than that of smartphones of the moment. Thus compart routing tables and routing information are a key research issue for macro mobility management. Second, which wireless communication is used for the device discovery is an important issue. A straightforward solution is a paging mechanism provided by a cellular network, but this wastes cellular frequency resources. Thus non-cellular wireless communications need be used and the LTE-based device-to-device (D2D) communication mechanism called ProSe [16–18] is a good candidate because its communication ranges is longer than that of Wi-Fi and energy consumption of devices is lower than that of cellular communications. However, since the ProSe's communication is not so long as that of cellular communications, how to leverage such short range communications without increasing energy consumption of IoT gateways.

The details of macro mobility management and IoT gateway/relay UE discovery mechanisms are described in Chapters 4 and 5, respectively.

Chapter 4

A Routing-Based Mobility Management Scheme for IoT Devices in Wireless Mobile Networks

Internet of Things (IoT) devices, which have different characteristics in mobility and communication patterns from traditional mobile devices such as cellular phones, have come into existence as a new type of mobile devices. A strict mobility management scheme for providing highly mobile devices with seamless access is over-engineered for IoT devices' mobility management. This chapter revisits current mobility management schemes for wireless mobile networks based on identifier/locator separation. This chapter focuses on IoT communication patterns, and propose a new routing-based mobility scheme for them. Our scheme adopts routing information aggregation scheme using the Bloom Filter as a data structure to store routing information. The author clarifies the effectiveness of the proposed scheme in IoT environments with a large number of IoT devices, and discuss its deployment issues.

The remainder of this chapter is organized as follows: Section 4.1 presents the proposed scheme using Bloom Filters. Section 4.2 evaluates the proposed scheme, and the author discusses deployment issues in Section 4.3. Finally, Section 4.7 concludes this chapter.

4.1 Routing-based Mobility Management Scheme

4.1.1 Design Principles

Departing from identifier/locator separation, the author proposes a new routing-based mobility management scheme wherein routing information about non-aggregated IoT devices' identifiers is advertised in an intra-domain network. The author considers that such a scheme is feasible from the observations on IoT communications.

First, target IoT applications do not require that a packet delivery rate is nearly one since it can be expected that slight dropping data packets of the IoT applications such as an environment monitoring is not critical. The author adopts an aggregation scheme such that the number of device states maintained at routes is reduced by allowing a slight decrease of the packet delivery rates.

Next, the author assumes that the networks have hierarchical topologies like Internet Service Provider (ISP) networks, and hence the proposed scheme takes care of aggregating routing information toward upstream routers from downstream ones.

Finally, the author assumes that IoT devices have unique identifiers (e.g., IP addresses or names) to distinguish devices from each other. Management of IoT devices' mobility based on routing convergence times could not be a critical issue since IoT devices for sensing or monitoring applications need not change their locations frequently.

4.1.2 Bloom Filter Based Routing

Fig. 4.1 shows an overview of the proposed Bloom Filter based routing architecture. In the proposed scheme, a router has Bloom Filters on a per-port basis to store routing information associated with the IoT devices that may exist in the direction toward the port. The Bloom Filter employs an array of m bits, which are initialized to all 0, and a fixed number k of hash functions $h_1(), h_2(), \dots h_k()$ which return integer values in the range from 1 to m [11]. All routers use the same filter size m and same k hash functions. The edge routers (i.e., R4 through R7 in Fig. 4.1) initially obtain routing information of the IoT devices in each edge networks.

Assuming that the router knows which ports upstream/downstream are, the router aggregates the routing information toward upstream routers periodically by the bitwise OR operation of its Bloom Filters and send the aggregated filter to the upstream router. For example, in Fig. 4.1, the router R2 aggregates its filters F_{21} and F_{22}, then sends the aggregated filter as F_{11} to the router R1. When a data packet, which an identifier of a destination device is X, arrives a router, the router tests a membership

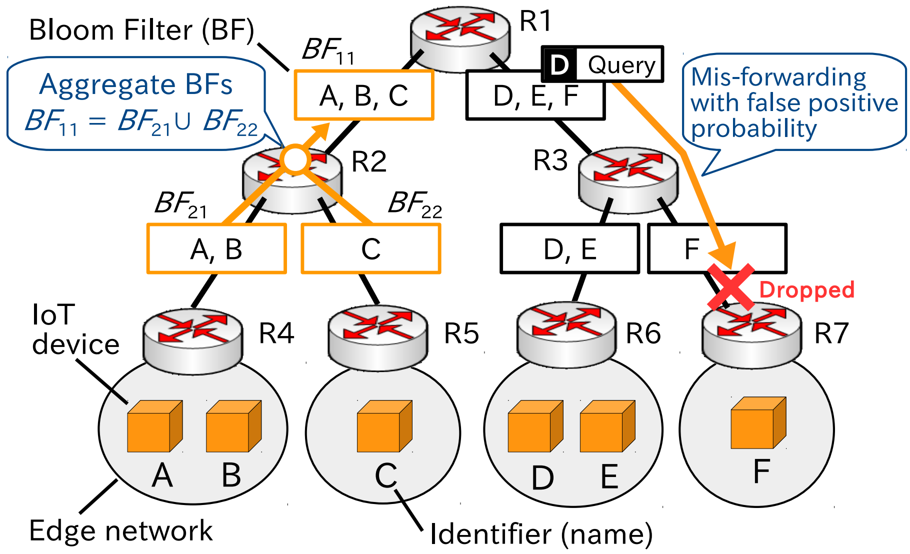


Figure 4.1: An overview of the proposed routing-based mobility management scheme: routers store routing information using the Bloom Filter data structure

of the identifier X in each their filters to decide the next hop. In the case when the router cannot find the next hop, it drops the data packet.

4.2 Performance Evaluation

This section evaluates the performance of the proposed mobility management scheme and investigates the effectiveness of size reduction of Bloom Filters through simulations.

4.2.1 Setup

The simulations assume an IoT environment wherein 500,000 IoT devices move in a square field independently. A mobility model of the IoT devices is truncated Lévy walk [40] describing human mobility because the author assumes that many devices are wearable devices. The parameters of the mobility model are shown in Table 4.1. The square field size is 512×512 [km^2], which is similar to the area of the main island of Japan or the United Kingdom. It is divided into 16,384 small square blocks, each size of which is 4×4 [km^2]. This size is similar to the area of a TA in cellular networks. A network topology consisting of routers is a complete quadtree of which height is seven. Edge routers, which locate at the leaves of the quadtree, manage the information of IoT devices in each small 4×4 [km^2] square block.

Table 4.1: Parameters for truncated Lévy walk model

Parameter	Value
flight scale	10
flight alpha	0.5
flight limit [m]	10000
pausing scale	0.1
pausing alpha	1.5
pausing limit [sec]	100
speed coefficient	18.72
speed power	0.79

The total simulation time is set to 3,600 seconds. The routers update routing information every 600 seconds. Queries to deliver data packets are generated every 50 seconds, and they will be routed to a randomly selected IoT device from the root router. Under the condition, the simulation evaluates the packet delivery rate.

4.2.2 Initial Results

Figure 4.2 shows the relation between the size of Bloom Filters m and the packet delivery rate. Fig. 4.2 indicates that the packet delivery rate takes around 0.9 in the case that the size of Bloom Filters m is larger than 0.6×10^6 [bit] ($\approx$ 73 [kbyte]). In contrast, in the case that the size of Bloom Filters m is smaller than 0.6×10^6 [bit], the packet delivery rate takes less than 0.9 since the false positive probability increases in the case that the size of the Bloom Filter is small. Such undelivered packets are not due to false positives of the Bloom Filters, but to the periodical update interval. When an IoT device moves before the next interval, packets to such moving IoT devices are dropped until the next interval. However, packets will be correctly delivered to the destination IoT device after the next interval.

This result suggests that the proposed scheme has the significant effectiveness of size reduction of Bloom Filters. For instance, if routing tables have one routing entry per destination, of which size is 24 bytes, the size of the routing table is approximately 11.4 Mbytes to store routing information of 500,000 IoT devices at a router. Since the author expects that the effectiveness of size reduction depends on the number of IoT devices or the mobility of IoT devices, it is important to more minutely evaluate the proposed scheme in order to set the parameter associated with real IoT environments accurately.

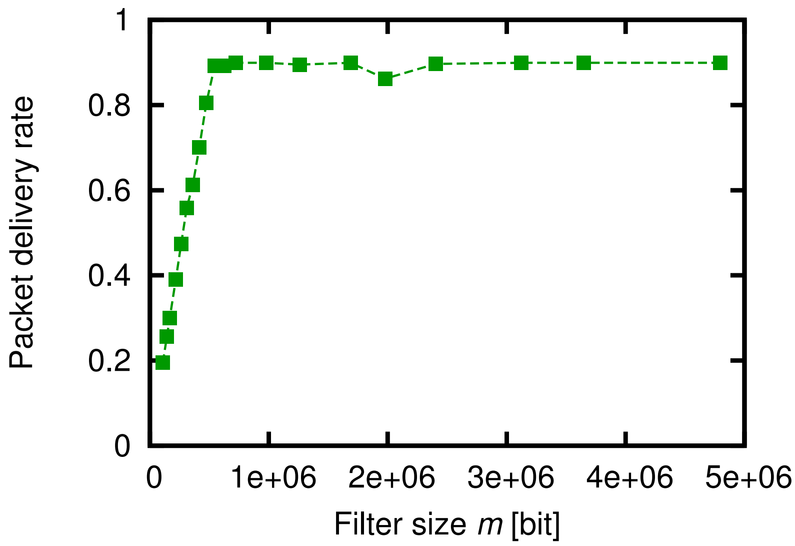


Figure 4.2: The relation between the size of Bloom Filters m [bit] and the packet delivery rate (simulation time [sec]: 3,600, number of devices: 500,000, mobility model: truncated Lévy walk, route update interval [sec]: 600)

4.3 Deployment Issues

This section presents deployment issues to show a migration path.

4.3.1 Handling Micro Mobility

This subsection addresses how the proposed routing-based scheme handles micro mobility. The word “micro mobility” means either that an IoT device moves within an area managed by an attachment point or that it moves to another area managed by a neighboring attachment point. For example, in the case of LTE, an eNodeB is an attachment point and it accommodates multiple cells as its areas [27].

The scheme hides the two cases of micro mobility in the following way: First, micro mobility in the same attachment point is inherently solved by placing a router at all attachment points, e.g., eNobeBs. Second, when an IoT device moves to another attachment point as discussed in Section 4.2, a packet to it is dropped. Thus the proposed routing-based scheme cannot inherently achieve 100% packet delivery ratio. However, 100% packet delivery is feasible just adding any of established micro mobility techniques to the proposed scheme. For instance, an LTE network handles such attachment point changes as follows: When downlink packets arrive at a source eNodeB during the handover procedures to a target eNodeB, the source eNodeB forwards these packets to the target eNodeB [27].

This forwarding mechanism can be easily adapted and built into an edge router of the proposed scheme.

4.3.2 Naming Scheme

Since the proposed scheme uses only IoT devices' identifiers, i.e., their names, for their mobility management, the choice of naming schemes is a crucial issue. The proposed scheme can support both flat and hierarchical names since the proposed scheme does not have any constraint on identifiers. However, the difference of the naming schemes has an impact on macro mobility. The word "macro mobility" means that a packet is routed to a gateway between a mobile wireless network which is controlled by the proposed scheme and external networks. Thus, the author addresses the following research question: Which naming scheme is better suited for the proposed scheme?

When the flat name is used, IoT devices have unique identifiers such as GUIDs [31]. In this case, all names of IoT devices should be advertised to routers in the mobile wireless network as well as to those in external networks. This means that the proposed scheme or another mobility management scheme provide scalability at the Internet scale, but that this achievement is still an open research question.

On the contrary, when the hierarchical name is used, some upper components of the name is used as a prefix as in NDN/CCN [41]. In this case, only such prefixes need be advertised to external networks and apparently, there is no scalability problem. Thus the proposed scheme prefers hierarchical names to flat ones.

4.3.3 Applicability to Large-Scale Networks

The proposed scheme can be built into existing large-scale wireless mobile networks such as LTE-based cellular networks. The scheme uses all the mechanisms except for forwarding/routing without any modification. For example, tracking and paging mechanisms of cellular networks are used as they are. Besides, co-existence with traditional mobility strict mobility mechanisms for seamless access is feasible by using router virtualization techniques.

This chapter uses a distance vector routing protocol on tree-like topologies because traditional sink-client communications, wherein information pieces are uploaded from devices to a single sink, are popular. However, the proposed Bloom Filter-based forwarding/routing mechanism is easily built into a link-state routing protocol, e.g., open shortest path first (OSPF). This adoption enables it for the proposed mechanism to support multiple sinks in large-scale non-tree topology networks. In the case of OSPF, router link state advertisements (LSAs) are represented as Bloom Filters of all devices' identifiers which a router accommodates. The router LSAs are flooded to all routers in a network and each router calculates the shortest path to the set of identifiers included in each router LSA. This enables it for each router and device to send a packet to anyone and thus multiple sinks can be supported in arbitrary topologies. In addition, the adoption of the link-state routing protocol makes it possible for the proposed mechanism to handle movements of the both sinks and IoT devices since it regards both root/sink nodes and IoT devices as devices attached to a network. This can achieve similar performances for communications from/to sinks in movement as shown in Section 4.2.

The link-state routing protocol naturally resolves a scalability issue as well. The network is divided into multiple OSPF stub areas, and a summary LSA, which is obtained by the bitwise OR operation of all router LSAs in a stub area, is advertised by an area border router to the others. Thus, each area border router has the Bloom Filters corresponding to summary LSAs, of which the number is the same as that of all stub areas. The author considers that this protocol is enough scalable to a large number of devices as described below. The author assumes that a cellular network accommodates 10,000,000 IoT devices and that the network is divided into 20 stub areas. In this case, each stub area accommodates 500,000 IoT devices and it is similar to the network in Section 4.2. Thus each area border router has only to have 20 Bloom Filters of 73 [kbyte] to support movement of the IoT devices as described in Section 4.2.

The details of the extension to OSPF is described in Sctiob 4.4.

4.4 Extension to OSPF

4.4.1 Protocol Design

In this section, the author extends a link state routing protocol, i.e., OSPF, to use Bloom Filters, as routing information, i.e., LSAs, whereas the routing protocol designed in Section 4.1 is based on a distance vector routing protocol. Goals of the extension are summarized below: First, arbitrary topologies are supported, whereas only tree topologies are supported by the distance vector routing protocol in Section 4.1. Second, the extension to OSPF enables for a user and an IoT device to communicate directly without via the root of the cellular network, i.e., P-GW. Besides the two level routing propagation in edge areas and a backbone area reduces the number of messages conveying routing information.

The author extends OSPF, i.e., a link-state routing protocol, to advertise a large number of IoT devices' identifies which are not aggregated. Since identifiers of IoT devices are assigned

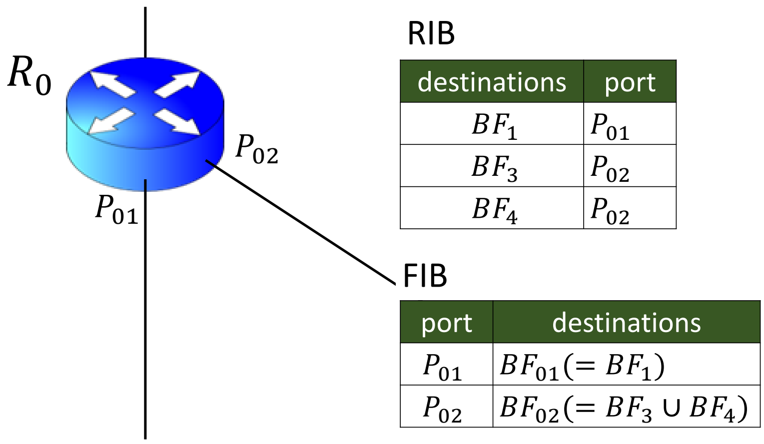


Figure 4.3: An overview of FIB

independently of their topological locations, it is difficult to aggregate even identifiers of IoT devices in a topologically adjacent area. In the similar way, the author also adopts a as a routing information piece to reduce its size. A router LSA is extended so that prefixes accommodated by a router are represented by a Bloom Filter. Precisely speaking, a router LSA consists of an identifier of a router and a Bloom Filter which stores accommodated IoT devices' identifiers as its members.

A router uses two data structures: a Routing Information Base (RIB) and a FIB. A RIB is used to recorded routing information obtained from received router LSAs as shown in Figure 4.3. An entry of a RIB is created by each router LSA and it consists of a pair a destination represented by a Bloom Filter and an output port. The Bloom Filter stores identifiers of all destination IoT devices. For example, Bloom Filter BF_1 for Router R_0's output port P_{01} stores IoT devices' identifiers A, B and C. On the contrary, a FIB is looked up to search an output port to which a data packet is forwarded. An entry of a FIB is a pair of an output port and destinations represented by a Bloom Filter. This Bloom Filter is obtained by a bitwise OR operation of Bloom Filters of the destinations in a RIB. It means that the Bloom Filter at the output port contains all the destinations. For example, two Bloom Filters BF_3 and BF_4 at output port P_{02} are bitwise ORed to Bloom Filter BF_{02}.

In the rest of the subsection, a walk through scenario of advertising router LSAs is explained using Figure 4.4. First, an edge router is responsible for IoT device registration and advertisement of accommodated IoT devices. When a new IoT device moves to a cell, it sends a location update message to an edge router via an eNB. The edge router adds the identifier to a Bloom Filter, i.e., a router LSA advertised by it. It periodically advertises a router LSA which have all the accommodated IoT devices' identifiers as members to all routers in an OSPF area. Such periodical advertisement is

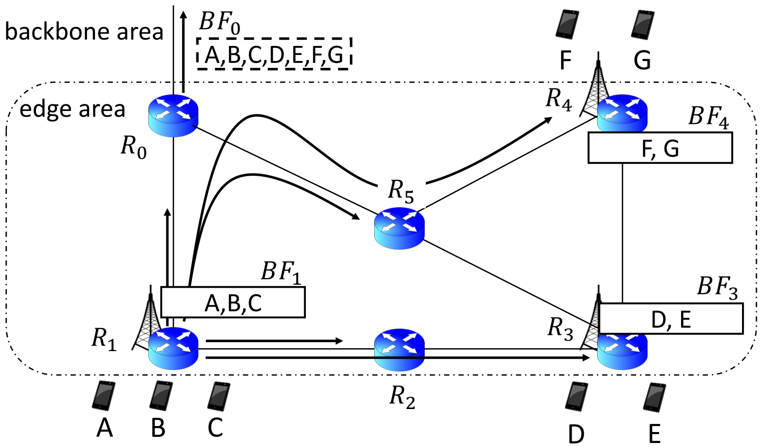


Figure 4.4: An overview of advertisement of router LSAs

adopted rather than on-demand advertisement such that a router LSA is advertised every when a new IoT device moves to/from a cell. This is because the objective is to reduce the number of router LSA advertisements. This router LSA is flooded to all the routers in an OSPF area. For example, router R_1advertises a router LSA which contains the identifiers of three IoT devices, i.e., A, B and C, to routers R_0, R_2,R_3,R_4 and R_5.

Second, an Area Border Router (ABR) advertises all IoT devices' identifiers to other OSPF areas via a backbone area. Precisely speaking, an ABR aggregates all Bloom Filters which are received from all routers in an edge area to a new Bloom Filter. The aggregation is performed with a bitwise OR operation of Bloom Filters. The Bloom Filter is a summary LSA of this area and the summary LSA with this Bloom Filter is advertised to all the ABRs in the backbone area. For example, in Figure 4.4, ABR R_0 advertises a router LSA containing Bloom Filter BF_0, which aggregates Bloom Filters BF_1, BF_3 and BF_4, to all the ABRs in a backbone area.

4.5 Routing-based Mobility Management vs Anchor-based Mobility Management

4.5.1 Overview

In Section 3, the author claims that the routing-based mobility management is better than the anchor-based one which current cellular networks adopt in terms of latency and traffic concentration. This section solves the large size of routing information by leveraging Bloom Filters. The remaining issue is control overhead. This section compares the numbers of control messages for the both management schemes.

4.5.2 Control Messages of Tunnel-based Mobility Management

The anchor-based mobility management scheme adopted by cellular networks, i.e., LTE networks, use a Mobility Management Entity (MME) as an anchor. The MME manages mappings of identifiers, i.e., home IP addresses, and current locations, i.e., care-of IP addresses. When a device moves to another cell, an update message containing a new mapping is sent to the MME and the mapping table at the MME is updated. In other words, the central server, i.e., the MME, manages all the mappings.

On the contrary, edge routers manage such mapping rather than the anchor in the routing-based mobility management scheme. When a device moves to another cell, an edge router next to the new cell advertises a routing message, i.e., a router LSA which has a new mapping to all routers in a cellular network. However, if a router LSA is advertised every when a cell moves from one cell to another, the number of router LSAs becomes large. In order to circumvent the problem, the thesis takes an approach of periodical advertisement as described in Section 4.1.2 with at the slight expense of the packet delivery rate.

This section compares the numbers of control messages, i.e., that of router LSAs in the routing-based mobility management scheme and that of (mapping) update messages in the anchor-based mobility management.

4.5.3 Control Messages of Anchor-based Mobility Management

An update message is sent from an eNB to the MME and from the MME to the Serving Gateway (S-GW). It is sent either (1) when a device moves one cell from another, or (2) when a state of device changes. How an update message is sent depends on a state of a device. If a state of a device is the idle state, the state changes to the connected state when it either sends or receives a packet. If a device does not either send or receive packet for a predetermined duration, the state changes to the idle state. The states of devices are managed by an MME which is responsible for a tracking areas. A tracking area consists of multiple cells. Table 4.2 summaries the number of control messages for each case.

(1) Control Messages Triggered by Device Movement

When a device of the connected state moves either from one cell to another, or from one tracking area to another, control messages are sent as follows: from the new eNB to the old eNB, from the

Table 4.2: The number of control messages in the Anchor-based Mobility Management

Event which Trigers an update	State of Device at the event	The number of Cotrol Messages
Movement	Connected	6
	Idle	2
State Change	Connected	5
	Idle	5 (Without Tracking)
		N_{TAcell} + 5 (With Tracking)

old eNB cell to new eNB, from the MME and the S-GW, from the S-GW to the MME. The number control messages is 6. On the contrary, when a device of the idle state moves from one tracking are to another, a control message is exchanged between the device and the MME. Thus the number of control messages is 2.

(2) Control Messages Triggered by Device's State Change

An eNB triggers change of a device's state from the idle state to the connected one. When a device does not either send or receive any packet for a predefined duration, an eNB exchanges 3 control messages with the MME and the MME and S-GW exchange 2 control messages. Thus the number of control messages is 5.

On the contrary, either an eNB or a device triggers change of a device' state from the idle state to the connected one. First. when a device at the idle state send a packet, it sends a control message to the MME and then the MME sends exchanges two control messages with both the eNB and the S-GW. Thus the number of control messages is 5. Second, when a device at the idle state receives a packet, the MME triggers discovery of this device. Since this device is not recorded in the mapping table, the MME sends control messages to all devices in the tracking area, which is called *paging*. Assuming that the number of all devices is N_{TAcell}, N_{TAcell} control messages are sent. In addition, five control messages are sent like the first case.

4.5.4 Control Messages of Routing-based Mobility Management

The proposed routing management adopts periodical advertisement. Thus control messages include router LSAs advertised by an edge router and control messages exchanged between a device a eNB, i.e., an edge router which the device moves to. Assuming that a router LSA is flooded to all routers in a cellular network, that the interval of periodical advertisement is T and that the number of routers is N_{router}, $2N_{router}(N_{router} - 1)$ router LSAs are sent at every interval. In addition, every when a device moves from one cell to another, two control messages are exchanged between the old and new eNBs.

4.6 Evaluation

This section evaluates feasibility of the proposed routing-based mobility management in terms of control message overhead and forwarding speed.

4.6.1 Control Messages

Setup

The simulations assume an IoT environment wherein 160,000 IoT devices move in a square field independently. A mobility model of the IoT devices is truncated Lévy walk [40] describing human mobility because the author assumes that many devices are wearable devices. The parameters of the mobility model are shown in Table 4.1. The square field size is 8×8 [km^2], It is divided into 16 small square blocks, each size of which is 2×2 [km^2]. This size is similar to the area of a tracking area in cellular networks and the number of cells in a tracking area is 4. The router topoloy is the complete quad-tree of which depth is 2 as illustraetd in Fig. 4.5. The P-GW is placed at the root router. The size of router LSA, i.e., the Bloom Filter, is 60 Kbytes.

The area structure of OSPF is designed as follows: All the routers comprise a single OSPF domain and the root router works as a border router (BR) to other domains, as illustrated in Figure reffig:ospf-area. The BR uses either e-BGP or i-BGP. The OSPF domain consists of 4 edge areas and each edge area consists of the 4 second level routers. Each first level router works as an area border router (ABR) for a subtree of the second level routers. The root router, second level routers and the third level routers comprise a backbone area of OSPF. The edge routers and ABRs advertise a router LSA and a summary LSA every routing update interval time, respectively.

The total simulation time is set to 54,000 seconds. The routing update interval in the backbone area is set to 3000 seconds, and that in stub areas is set to 600 seconds. Hosts in the Internet sends packets all IoT devices in the cellular network and packets are sent accordong to Poission process.

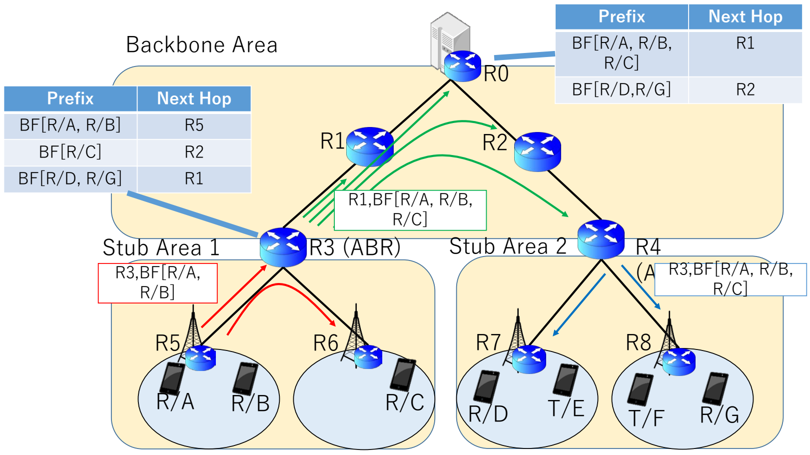


Figure 4.5: Stub Areas and Backbone Area

Results

Figure 4.6 shows the numbers of control messages for the both management schemes. The following observations are obtained from the simulation results. The number of the routing-based mobility management is smaller than that of the anchor-based one. This is because devices at the idle state send control messages to update their locations in the anchor-based management. Especially, control messages due devices' state changes are dominant because paging in a tracking area frequently occurs. In conclusion, the routing-based management avoids such control messages due to paging by recording devices' states at edge routers.

4.6.2 Packet Forwarding Speed

Since FIB lookup is heavy computation that might degrade packet forwarding performances, we measure CPU cycles of FIB lookup in order to confirm that FIB lookup based on the proposed Bloom Filters does not degrade packet forwarding performance We measure FIB lookup cycles on the following configurations: First, as a hardware platform, we use a computer having two Xeon X5570 CPU (2.93 GHz $\times$ 4 CPU cores). The operating system on the computer is Ubuntu 10.04 LTS. We use one CPU core for measurements. Second, we measure CPU cycles spent for FIB lookup with the read time-stamp counter (RDTSC) instruction, which reads the time stamp counter, i.e., a register incremented by CPU cycles. We compute CPU cycles of FIB lookup by adding the RDTSC instruction at the beginning and end of the FIB lookup source code. Third, a router is assumed to have five output ports and thus the measured FIB lookup cycles are a sum of those for the five output

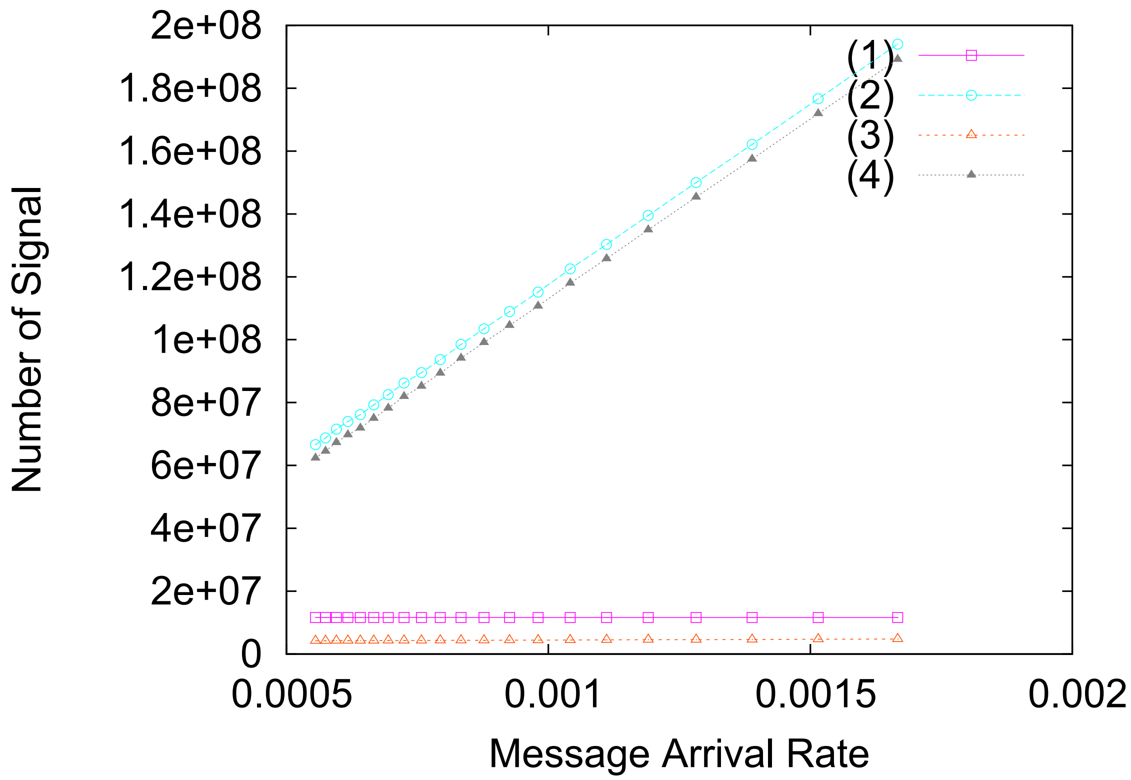


Figure 4.6: Comparison of Tunnel-based and Routing-based Management Schemes
((1)the total number of control messages of routing-based manegement (2)the total number of control messages of anchor-based management (3)the number of control messages due to devices' movement of anchor-based management (4)the number of control messages due to devices' state changes of anchor-based management)

ports.

The measurement shows that CPU cycles of FIB lookup is 2,401.17 when the size of Bloom Filters is 9.8×10^6 [bit] with 7 hash functions. The result suggests that FIB lookup based on the proposed Bloom Filters is light-weight.

4.7 Conclusion

This chapter has proposed a routing-based mobility management scheme using Bloom Filters for IoT devices, and has clarified the effectiveness of the proposed scheme. The author has shown that the proposed scheme can reduce the size of routing information, which are stored in Bloom Filters at routers, to approximately 73 kbytes in the case that required packet delivery rate is relaxed to around 0.9. The contributions of this chapter are summarized as below:

- This chapter designs a link-state routing protocol by extending OSPF based on the proposed Bloom Filters as routing information. The simulations show that the proposed routing-based mobility management provides scalability needed for a regional cellular network.

- The routing-based mobility management provides smaller overhead than anchor-based mobility management which current cellular networks adopt.

Chapter 5

Relay Mobile Device Discovery with Proximity Services for User-provided IoT Networks

Internet of Things (IoT) devices deployed in urban areas are seen as data sources for urban sensing IoT applications. Since installing cellular interfaces on a huge number of IoT devices is expensive, the author proposes to use a user equipment (UE) device with a local wireless interface as a mobile IoT gateway for fixed IoT devices. This chapter designs a new mobile architecture based on cellular networks to accommodate non-cellular fixed IoT devices by UE devices working as IoT gateways. One key feature is that the proposed architecture leverages proximity services (ProSe) to discover relay UE devices with low overhead in terms of discovery messages. Simulation studies clarify the feasibility of the proposed architecture including the relay UE discovery mechanism in urban areas.

The rest of this chapter is organized as follows. Section 3.1 explains a target IoT application. Section 3.3 explains our mobile architecture and relay UE discovery mechanisms. Section 5.2 presents the simulation results, and Section 5.3 discusses the feasibility and implementation issues of the proposed architecture and discovery mechanism. Finally, Section 5.4 concludes this chapter.

5.1 Proposed Mobile Architecture

5.1.1 Assumptions

A proposed mobile architecture is based on the LTE architecture. Assumptions on a network environment are summarized as follows:

- IoT devices *only* have a Wi-Fi interface, and thus UE devices of LTE, i.e., relay UE devices, relay packets between IoT devices and hosts in the Internet.

- Wi-Fi Aware [42] and Wi-Fi Direct [43] are assumed to be used for communication between an IoT device and a relay UE device. Please note that Wi-Fi Aware is used only when a relay UE device establishes a connection to an IoT device.

- The IoT device always selects UE devices in its Wi-Fi range as a relay UE device. Once an IoT device establishes a connection to a relay UE device, the connection is sustained until the relay UE device moves away from the Wi-Fi range of the IoT device.

- IoT devices and evolved node Bs (eNodeBs) provide ProSe functions for relay UE discoveries.

- For reducing energy consumption of relay UE devices, if a certain period of time, hereinafter reffered to as *ProSe idle time*, has passed since the IoT device disconnected from its last relay UE device, it disables its ProSe functions until it connects to a new relay UE device.

5.1.2 Naming Scheme

The author adopts NDN as a naming and forwarding scheme for IoT devices because host-based scheme like IP uses a resolution system from a device's identifier to its physical address like domain name system (DNS). This name resolution incurs additional round trip time before requesting a sensed data piece to an IoT device. Besides, NDN naturally supports IoT devices' mobility without changing their names by leveraging name-based routing/forwarding. The author uses a routing-based mobility management [10, 44], wherein names of IoT devices are recorded by NDN routers in eNodeBs and they are advertised to all routers in a cellular network. Bloom Filters are used to aggregate non-aggregatable IoT devices' names.

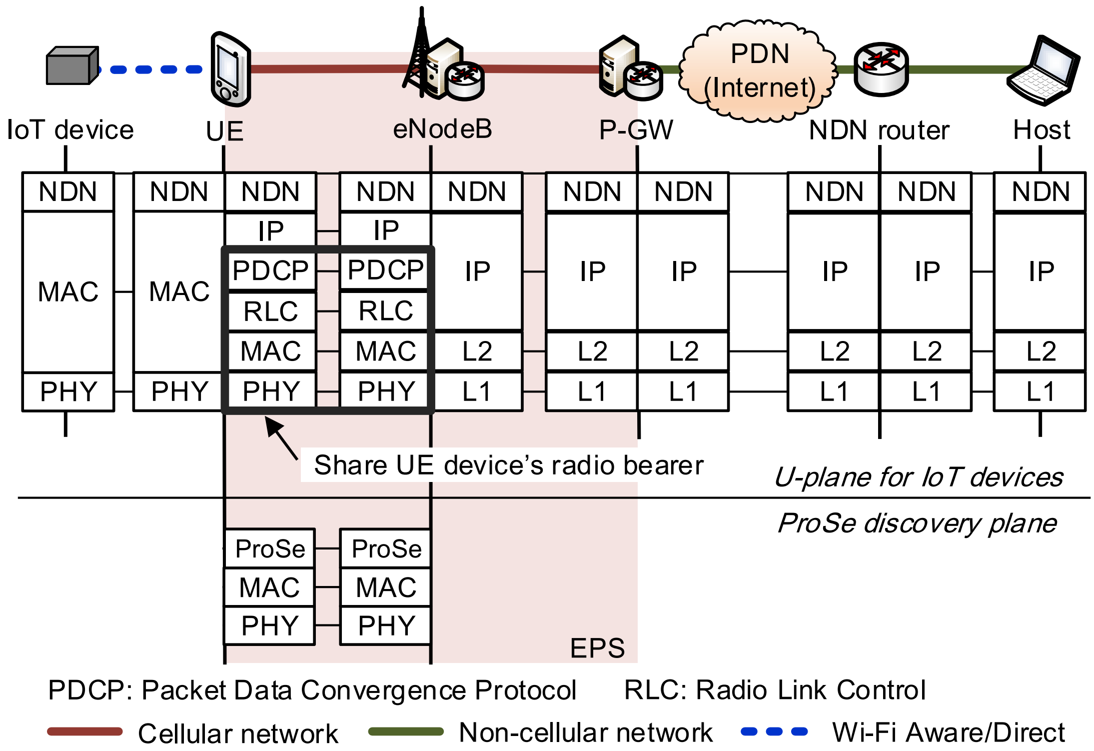


Figure 5.1: Protocol stack

5.1.3 Protocol Stack

The protocol stack is shown in Fig. 5.1. A packet core network, i.e., an EPC network is overlaid by a NDN network which is dedicatedly used for IoT devices' communications. NDN router's functionalities are added to relay UE devices, packet data network gateways (P-GWs) and eNodeBs. This means that relay UE devices, P-GWs and eNodeBs are NDN-capable. Hereafter, a pair of a P-GW and an NDN router is simply reffered to as a P-GW and a pair of an eNodeB and an NDN router is reffered to as an eNodeB. Other local IP addresses in the EPC network are assigned to eNodeBs than those for PDN connections to establish IP tunnels between NDN faces of P-GWs and eNodeBs. The eNodeBs behave as the P-GW proxies for IoT devices. P-GWs distinguish NDN packets from other IP packets associated with UE devices' PDN connections. In a radio access network, i.e., an E-UTRAN network, a connection for the NDN-based IoT application and a relay UE device's PDN connection share the same radio bearer between an eNodeB and the relay UE device. This is because using dedicated radio bearers for transmitting NDN packets sent from/to IoT devices incurs the large overhead of bearer management. Mobility of relay UE devices is handled by the mobility management protocol of the EPC, and that of IoT devices is handled by a routing protocol of the overlaid NDN protocol. Precisely speaking, IoT devices' names are periodically advertised to forwarding information bases (FIBs) of relay UE devices, NDN functionalities of eNodeBs and P-GWs. Their entry called FIB entry, which consists of an IoT device's name prefix, faces, and its expiration time, is soft-state to reduce message overhead for updating FIBs.

Please note that IoT devices are stationary and thus relay UE devices for some IoT device are always connected to the same eNodeB. In other words, any interest packet to such an IoT device is always forwarded to the same eNodeB to which its potential relay UE devices are connected.

5.1.4 Forwarding

Forwarding an interest message to destination IoT devices is a key role of the proposed architecture. This is because a key use case of the proposed architecture is that pieces of sensed data stored in an IoT device are retrieved from application users' hosts in the Internet. The interest message is forwarded from the host to the IoT device via the relay UE device as follows. First, the interest message from the host is forwarded to a P-GW assuming that prefixes which aggregate IoT devices' names are advertised to the Internet periodically. Second, the interest message is forwarded to the eNodeB through the P-GW using the name-based routing of NDN. Since IoT devices are usually fixed, their names are stored in FIBs of eNodeBs. Thus forwarding the interest message to the eNodeB is trivial. Finally, the eNodeB should forward the message to the relay UE device which is connected to the IoT devices. However, this forwarding is *not* trivial because recent relay UE devices may not be current relay ones due to their mobility and short range of Wi-Fi communication.

One of the key issues is how eNodeBs discover the relay UE device. In the rest of this section, the author explains the ProSe services and describes a relay UE discovery mechanism using the *ProSe Direct Discovery (Model B)*.

5.1.5 Overview of ProSe

The ProSe mainly provides functions: D2D device discovery, i.e., ProSe-enabled UE discovery, and D2D communication [16, 18]. In the *direct discovery* process, an UE device directly discovers nearby UE devices using its ProSe-enabled cellular interface, and a transmission range of its ProSe communication range for the discovery is summarized below. The maximum transmission range is approximately 1,000 [m] [45] when the D2D communications use all cellular frequency resources. On the contrary, the feasible transmission range is about 400 [m] as described below. Thus the

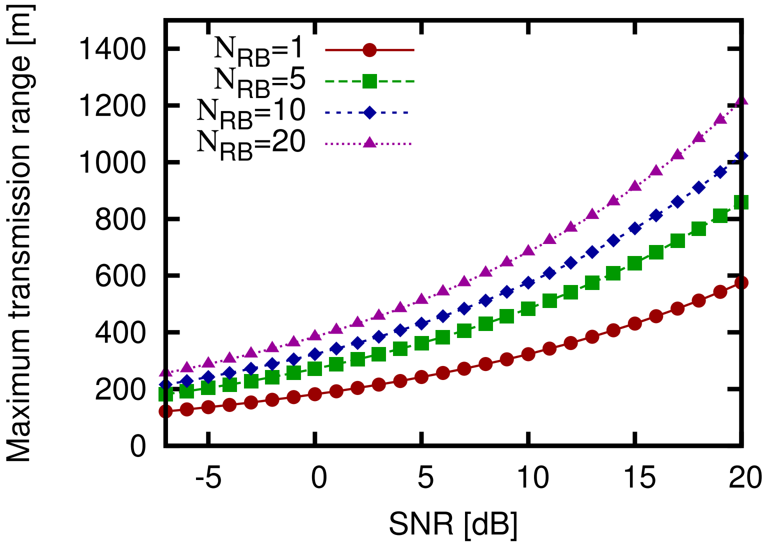


Figure 5.2: The relation between a signal-to-noise ratio (SNR) and the maximum transmission range of D2D [m]. (N_{RB}: the number of consumed resource blocks for physical transmission)

transmission ranges take intermediate values between these of the Wi-Fi and cellular interfaces, and this longer range than that of Wi-Fi enables the proposed architecture to perform a partial relay UE discovery in a portion of a macro cell.

The author addresses an issue of the maximum radius used by the proposed D2D-based discovery mechanism because the maximum transmission power of D2D P_{tx} is 23 [dBm], which is 10 or 20 [dBm] smaller than typical downlink transmission power.

To clarify the maximum cell radius, the author first investigates the maximum transmission range of D2D with consideration of the transmission power regulation. In this chapter, to calculate the transmission range, the author uses the system model of D2D communication in LTE environment proposed in [46]. The author assumes an LTE base station deployed at the center of a cell, and set its bandwidth = 5 [MHz], P_{tx} = 23 [dBm], and a noise power P_N = -174 [dBm]. The author uses the same mapping function between signal-to-noise ratio (SNR) values and modulation and coding scheme (MCS) indexes according to [46].

Figure 5.2 shows the relation between the signal-to-noise ratio (SNR) and the maximum transmission range of D2D in meters in the case that the number of consumed resource blocks for physical transmission N_{RB} is in the range of 1–20. Fig. 5.2 indicates that the maximum transmission range is approximately 400 [m] in the case that radio signal quality is medium, i.e., the value of the SNR is in the range of 5–10. In the case that radio signal quality is good, i.e., the value of the SNR is

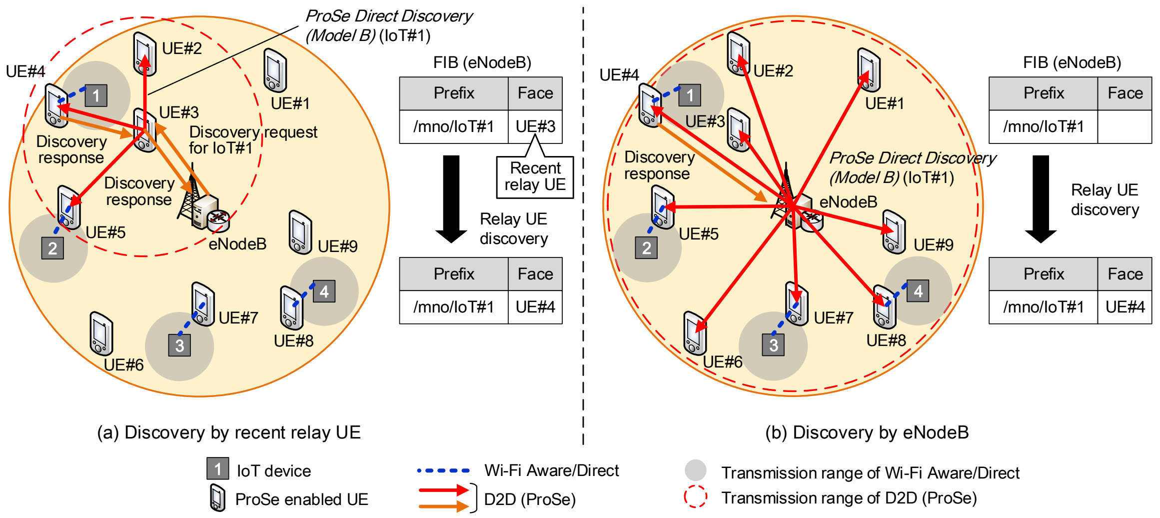


Figure 5.3: Two types of ProSe-assisted relay UE discovery mechanism. (target IoT device: IoT#1, recent relay UE device in the case (a): UE#3)

more than approximately 17, the maximum range can be set to 1,000 [m] if the D2D communications enable to consume more than 10 RBs for physical transmission.

The author also assumes that the D2D-based relay UE discovery mechanism uses dedicated D2D carriers. This is because if the discovery mechanism uses shared radio resources with normal cellular communications, normal communications' performance degrades when relay UE discovery is performed. To prevent the performance degradation, the author uses operator controlled D2D communication technologies [47], which dynamically assign a dedicated radio resource pool to D2D connections among UE devices or between eNodeBs and relay UE devices.

5.1.6 ProSe-assisted Relay UE Discovery Mechanism

Overview

Relay UE discovery is performed by the eNodeB in the two steps as shown in Fig. 5.3: *Relay UE Discovery by Recent Relay UE* and *Relay UE Discovery by eNodeB*. This assumes that the recent relay UE device is probably recorded in the FIB of the eNodeB. The word "recent" means that the relay UE device is recorded in the FIB of the eNodeB. But, it does not guarantee that the recent UE device is still a relay UE device because it may move away and because a new relay UE device is not advertised to the eNodeB.

The discovery works as shown in Algorithm 1. First, if a recent relay UE device for a target IoT

Algorithm 1 Relay UE discovery in eNodeBs

Input: A name of a target IoT device: *name*,
An identifier of a recent relay UE device: *recentRelayUE*.
Output: An identifier of a new relay UE device: *relayUE*.

```
procedure RelayUEDiscovery(name, recentRelayUE)
    relayUE ← empty
    if recentRelayUE is not empty then
        if A name's face entry was updated
          in the past discovery unnecessary period then
            relayUE ← recentRelayUE
        else
            SendDiscoveryRequestToRecentRelayUE(name)
            /* The recent relay UE device performs the relay UE
            discovery, and sends discovery results to the eNodeB.*/
            relayUE
            ← ReceiveDiscoveryResultFromRecentRelayUE
               (recentRelayUE)
        end if
    end if
    if relayUE is empty then
        relayUE ← RelayUEDiscoveryByENodeB(name)
    end if
    recentRelayUE ← relayUE
    return relayUE
end procedure
```

device is recorded as a face entry in its FIB when an eNodeB receives an interest message, it checks the elapsed time since the FIB entry of the target IoT device is updated. If the time period is shorter than a certain time period, which is reffered to as *ProSe discovery unnecessary time*, it forwards the interest message to the recent relay UE device without performing the relay UE discovery process. This is because the relay UE device may not move away within the certain time period, so that the interest message may be delivered with high probability. Otherwise, the eNodeB forwards a relay UE device discovery request message to the recent relay UE device. The recent UE device asks neighboring UE devices in the range of ProSe service. If the recent relay UE device works still as the relay UE device, the interest message is just forwarded to the IoT device. If a new relay UE device is discovered, the interest message is forward to it. Second, if the discovery fails, the eNodeB asks all UE devices in the cell. Finally, after the eNodeB receives discovery results containing an identifier of a new relay UE device, the interest message is forwarded to the target IoT devices via the found new relay UE device. If the new relay UE device is not found, the interest message is dropped at the eNodeB. The detail of transmitting interest and data messages is described in Section 5.1.7.

Relay UE Discovery by Recent Relay UE

First, the *Relay UE Discovery by Recent Relay UE* process is explaind. If a recent relay UE device is recorded as a face entry in a FIB of an eNodeB when an interest packet to a target IoT device arrives at the eNodeB, the proposed architecture performs this relay UE discovery process. Figure 5.3a shows an overview of the discovery process.

This discovery process consists of the following steps:

1. First, the eNodeB sends a discovery request message with a content name of the interest packet to the recent relay UE device, which corresponds to the face.

2. When the recent relay UE device receives the request message, it performs the *ProSe Direct Discovery (Model B)* [18] process to find a new relay UE device in a ProSe transmission range of the recent relay UE device.

3. The new relay UE device replies a response message with its identifier to the recent relay UE device, and it then sends the response message to the eNodeB.

4. Finally, the eNodeB forwards the interest packet to the found new relay UE device, and updates its face entry. If the new relay UE device is not found, the eNodeB performs a *Relay UE Discovery by eNodeB* process, which is described in the next section, to discover a new relay UE device in the whole cell.

In Fig. 5.3a showing an example of retrieving data from the IoT device#1, recent and new relay UE devices are UE devices#3 and #4, respectively.

This relay UE discovery mechanism can reduce the number of searched UE devices, which receive discovery messages using the ProSe communications, since a recent relay UE device requested from an eNodeB discovers only in its ProSe transmission range, which is usually smaller than a cell radius. In other words, this partial relay UE discovery avoids broadcasting discovery messages to all of UE devices in the cell.

Relay UE Discovery by eNodeB

If a recent relay UE device is not recorded in a FIB of an eNodeB when an interest packet to a target IoT device arrives at the eNodeB, the proposed architecture performs this relay UE discovery process.

Figure 5.3b shows an overview of the *Relay UE Discovery by eNodeB* process.

This relay UE discovery process works as follows:

1. The eNodeB performs the *ProSe Direct Discovery (Model B)* process directly to find a new relay UE device in whole of a cell. Since the ProSe functions are installed in eNodeBs, the eNodeBs can behave as virtual ProSe-enabled UE devices whose transmission range is equal to a cell radius.

2. The new relay UE device replies a response message to the eNodeB.

3. Finally, the eNodeB forwards the interest packet to the found new relay UE device, and update its face entry of the target IoT device in the FIB.

In Fig. 5.3b showing an example of retrieving data from the IoT device#1, a new relay UE device is a relay UE device#4.

Since, in this relay UE discovery process, an eNodeB broadcasts discovery requests to all UE devices in a cell, it is hard to reduce power consumption of the UE devices, however, it is certainly able to find a new relay UE device with higher probability than if the *Relay UE Discovery by Recent Relay UE* mechanism is performed, that is, it can reduce the packet dropping probability.

5.1.7 Transmitting Message Sequence

Finally, an example of a message sequence between an application user's host and a target IoT devices is explained in this section.

Figure 5.4 shows a message sequence when a host obtains data from a target IoT device#1. Details on obtaining sensed data are summarized in the following steps:

1. The IoT device#1 establishes a Wi-Fi Aware connection to the relay UE device#2 when the relay UE device#2 enters a Wi-Fi communication range of the IoT device#1. Both the IoT device#1 and the relay UE device#2 then create a face.

2. The application user sends an interest packet with a content name of the IoT device#1 when he or she wants to obtain data sensed by the IoT device#1.

3. The interest packet arrives at a P-GW of a cellular network accommodating the IoT device#1, and is then forwarded to an eNodeB, which manages a cell accommodating the IoT device#1.

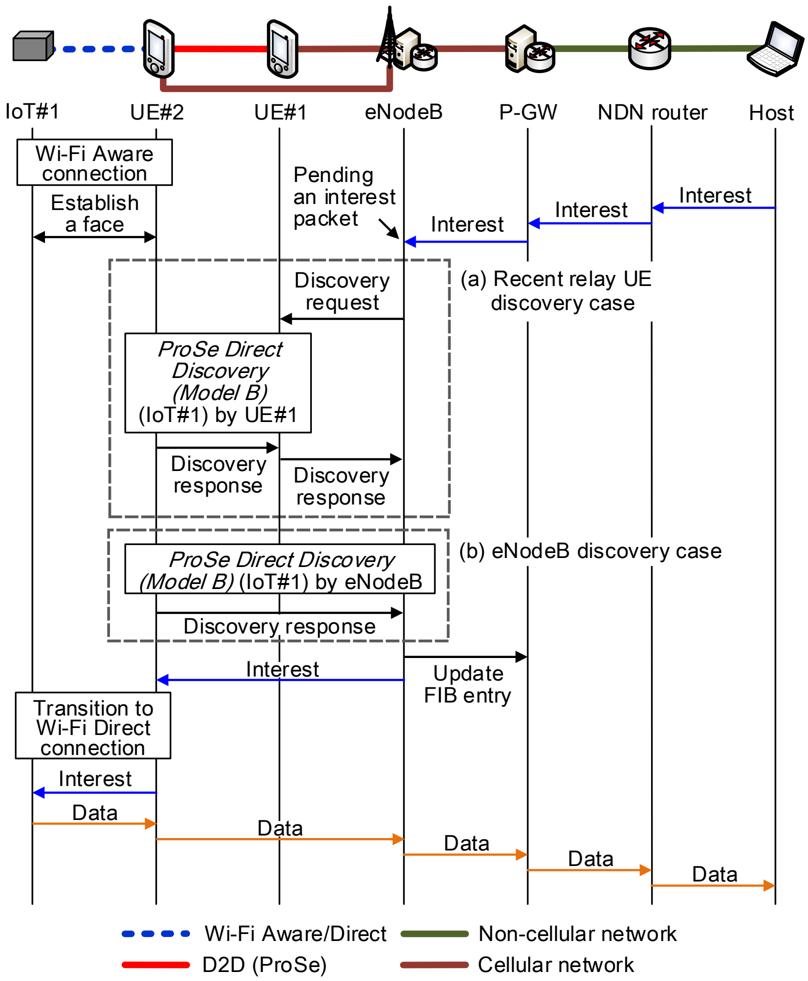


Figure 5.4: Message sequence to obtain data from an IoT device#1 via a relay UE device#2.

4. When the interest packet arrives at the eNodeB, the eNodeB discovers a relay UE device for the IoT device#1 using the *Relay UE Discovery by Recent Relay UE* or *Relay UE Discovery by eNodeB*, which are shown of cases (a) and (b) in Fig. 5.4 respectively, depending on existence of face information of the IoT device#1 in the eNodeB.

5. If UE device#1, which is a recent relay UE device in the case (a), fails to discover a new relay UE device, the *Relay UE Discovery by eNodeB* is also performed after the *Relay UE Discovery by Recent Relay UE* process.

6. After the relay UE discovery, the eNodeB forwards the interest packet to found a relay UE device#2. If it does not receive an identifier of the relay UE device, the interest packets are

Table 5.1: Parameters for truncated Lévy walk model

Parameter	Value
flight scale	10
flight alpha	0.5
flight limit [m]	100
pausing scale	0.1
pausing alpha	1.5
pausing limit [sec]	30
speed coefficient	18.72
speed power	0.79

dropped.

7. The relay UE device#2 then forwards the interest packet to the IoT device#1.

8. The IoT device#1 sends data packets to the application user's host as responses to the received interest packet.

5.2 Performance Evaluation

This section evaluates the performance of the proposed mobile architecture through simulations. In this chapter, the author focuses mainly on a wireless part of the proposed architecture because the architecture is vulnerable to the number of UE devices and their movement. To clarify the feasibility, the section evaluates the following metrics: packet delivery ratios between an eNodeB and IoT devices and relay UE discovery overheads. Results in the case that the number of allowed relay UE connections for an IoT device k is one are presented in Sections 5.2.2 and 5.2.3, and results in the case that k is more than one are presented in Section 5.2.4.

5.2.1 Simulation Conditions

In the simulations, 100 fixed IoT devices are deployed uniformly within a macro cell. The radius of the circular area is either 400 [m] or 1,000 [m], as described in Section 5.1.5.

According to the result in Fig. 5.2, when a sub-channel size in the frequency domain is 180 [kHz] and a SNR value is more than 17, 20 resource blocks (RBs) are consumed in 1 [ms] to gain the 1,000 [m] transmission range. The ratios of the consumed RBs to the total available RBs are 80, 40 and 20 [%] when bandwidths of the LTE system are 5, 10 and 20 [MHz], respectively. However, these large ratios are *not* critical issues since D2D communications for the discovery processes do not occur frequently. Thus, the author uses the radius of 400 [m] and that of 1,000 [m], assuming medium and good radio signal quality.

An IoT device has a Wi-Fi interface, of which communication range is 25 [m]. A ProSe-enabled UE device has both of cellular and Wi-Fi interfaces, and its ProSe communication range is 100 [m]. An expiration time of FIB entries at an eNodeB is 30 [sec] because it is assumed that it takes about 30 [sec] for persons with UE devices to come across the IoT devices' local communication coverage area. To allow UE devices' ProSe functions to enter the idle state for their energy saving, the ProSe idle time and the ProSe discovery unnecessary time are set to 5 and 20 seconds, respectively.

A mobility model of UE devices is truncated Lévy walk [40] describing human mobility because the author assumes that relay UE devices with walking persons in streets are able to keep a connection to IoT devices for a certain time to relay interest packets to the IoT devices. The parameters of the mobility model are shown in Table 5.1.

UE devices' movements and connections between the IoT devices and UE devices for 5,000 [sec] are simulated with a C++ event-driven simulator. Under the assumption that physical layer transmission in the E-UTRAN is successfully performed, the physical layer transmission is not modeled in the simulator. Since access patterns of actual web cameras and CCTVs usually have temporal locality [48], it is assumed that interest packets to each IoT device arrive at the eNodeB according to a Poisson process with rate λ [packets/sec]. To obtain one result, ten simulation trials are performed and the mean over the ten trials is calculated.

5.2.2 Packet Delivery Ratio

First, the author investigates the effects of the number of UE devices in a cell on the packet delivery ratio. Since one or more UE devices exist around an IoT device with high probability when the number of UE devices in a cell is large, it can be expected that the proposed architecture can achieve a high packet delivery ratio in a crowded environment such as urban areas. Figure 5.5 shows the relation between the number of UE devices in a cell and the packet delivery ratio when the Poisson process is used as the interest packet arrival process and the interest packet arrival rate λ is in the range of 0.03–10.0.

Fig. 5.5 indicates that when the number of UE devices in a cell is more than 10,000, the proposed architecture achieves a high packet delivery ratio ($\geq 95\%$). This result suggests that the proposed architecture is applicable to urban areas, which a certain number of UE devices usually exist around IoT devices. The architecture is suitable for environments where IoT devices are deployed at crowded spots in urban areas such as intersections and bus stops.

It is also observed that the difference of the interest packet arrival rates λ does not affect the packet delivery ratio significantly. In other words, the number of UE devices in a cell is a main factor of deciding the packet delivery ratio.

5.2.3 Relay UE Discovery Overhead

Next, the author investigates the effects of the number of UE devices on the relay UE discovery overhead. In this chapter, to evaluate the discovery overhead, the author uses the number of times an UE device is searched for an hour as a numerical metric. Figure 5.6 shows the relation between the number of UE devices in a cell and the number of times an UE device is searched for an hour when the Poisson process is used as the interest packet arrival process and the interest packet arrival rate λ is in the range of 0.03–10.0. The author chooses the rates by assuming environment monitoring applications, where one of IoT devices is searched every minute, every half an hour, or every hours. The range of frequencies corresponds to the range of λ from 1.60 to 0.03. In addition, the author chooses 10 of λ to know the overhead when a relay UE device is searched every several seconds.

Fig. 5.6 indicates that when the number of UE devices in a cell is more than 30,000, the proposed architecture reduces a frequency at which an UE device is searched by the recent UE devices or the eNodeB to less than about 45 [times/hour] when λ is in the range of 0.03–1.00. The frequency is 120 [times/hour] when λ is in the range of 2.00–10.0. 45 and 120 [times/hour] mean that each UE device is searched every 80 and 30 seconds on average, respectively. From this result, the ProSe-assisted relay UE mechanism has the significant effectiveness of reduction of the discovery overhead in the urban areas.

In Fig. 5.6(b), the curve at $\lambda = 0.03$ gets crossed with different curves. First, UE devices are searched more frequently when $\lambda = 0.03$ than when $\lambda = 0.10, 0.50$, and 1.00, when the number of UE devices in a cell is in the range of 1,000–10,000. This is because the number of recent relay UE devices in the FIB at $\lambda = 0.03$, which are used by partial discovery process, is smaller than those of the other cases due to combination of lower density of UE devices and a long inter-arrival time of interest packets. Second, UE devices are searched a little more frequently when $\lambda = 1.00$ than when $\lambda = 0.03$. This is because the frequency at relay UE discovery processes are performed when $\lambda = 0.03$ is simply less than that when $\lambda = 1.00$.

The role of (a) discovery by recent relay UE device is to reduce the number of discovery processes of (b) discovery by eNodeB. The simulation results show that the number of discovery processes of (a) discovery by recent relay UE device is larger than that of discovery processes of (b) discovery by

Table 5.2: Execution ratio r_{exec} and the number of searched UE devices per a discovery $\overline{N}_{\mathrm{SearchedUE}}$ of two types of relay UE discovery functions. (average, cell radius: 400 [m], the number of fixed IoT devices: 100, the number of UE devices in a cell: 20,000)

	(a) Discovery by recent relay UE device		(b) Discovery by eNodeB		(a) + (b)
λ [packets/sec]	r_{exec} [%]	$\overline{N}_{\mathrm{SearchedUE}}$	r_{exec} [%]	$\overline{N}_{\mathrm{SearchedUE}}$	$\overline{N}_{\mathrm{SearchedUE}}$
0.03	37.7	1.55	62.3	109.2	67.6
0.10	56.3	1.35	43.7	110.5	29.0
0.50	85.6	1.07	14.4	145.8	8.65
1.00	92.2	1.03	7.8	179.8	5.53
2.00	95.9	1.02	4.1	207.7	3.37
10.0	99.1	1.00	0.9	756.7	2.60

eNodeB. This contributes to reducing relay UE discovery overhead. For example, Tab. 5.2 shows the execution ratio r_{exec}, which is the ratio of the number of times each discovery process is performed, and the number of searched UE devices per a discovery process $\overline{N}_{\mathrm{SearchedUE}}$ of two types of relay UE discovery functions when the cell radius is 400 [m], the number of fixed IoT devices is 100, the number of UE devices in a cell is 20,000, $k = 1$, and λ is in the range of 0.03–10.0. This result implies the following observations:

- When λ is large, the number of discovery processes of (a) discovery by recent relay UE device is larger than that of discovery processes of (b) discovery by eNodeB.
- In any λ cases, the (a) discovery by recent relay UE device process certainly reduces the number of searched UE devices per its discovery process than that of the (b) discovery by eNodeB process significantly.

Thus this partial relay UE discovery process (a) is an important mechanism to reduce the discovery overhead.

These experimental results prove the feasibility of the proposed architecture in the urban areas.

5.2.4 Effects of the Number of Allowed Relay UE Connections

Finally, this section investigates the effects of the number of allowed relay UE connections on packet delivery ratios and the relay UE discovery overhead. Figures 5.7(a) and 5.7(b) show the relation between the number of UE devices in a cell and the packet delivery ratio and the relation between the number of UE devices in a cell and the number of times an UE device is searched for an hour, respectively, where the number of allowed relay UE connections k is in the range of 1–5. All other parameters remain the same as in Figs. 5.5 and 5.6.

Fig. 5.7(a) indicates that the difference of the number of allowed relay UE connections k does not affect the packet delivery ratio significantly. Meanwhile, Fig. 5.7(b) indicates that when the number of UE devices in a cell is more than 10,000 and k is more than one, the proposed architecture increases a frequency at which an UE device is searched than when the k is one. This is because the area searched by multiple relay UE devices in the *Relay UE Discovery by Recent Relay UE* process is larger than when a single relay UE device discovers.

From this result, the number of allowed relay UE connections k does not improve the performance of the proposed architecture significantly. Since allowing multiple relay UE connections causes wasteful consumption of relay UE device's battery, it is enough for the proposed architecture that k is one.

5.3 Discussion

5.3.1 Extension to Macro Cells

As shown in Section 5.1.5, the maximum cell radius which the proposed mechanism supports is about 400 [m] when the radio signal quality is medium.

There are a couple of approaches to enable the proposed architecture to be used in a macro cell of which radius is more than 400 [m] even when the radio signal quality is bad. First, a simple but effective approach is to place additional relay stations. For example, only several relay stations need be additionally placed in a macro cell of which radius is 1000 [m]. The second approach uses multiple cell-edge UE devices to discovery relay UE devices, which locate out of an eNodeB's D2D communication range. When an eNodeB performs the discovery process, it also sends a discovery request message to some cell-edge UE devices and these UE devices then perform the partial discovery process using their local communication interfaces.

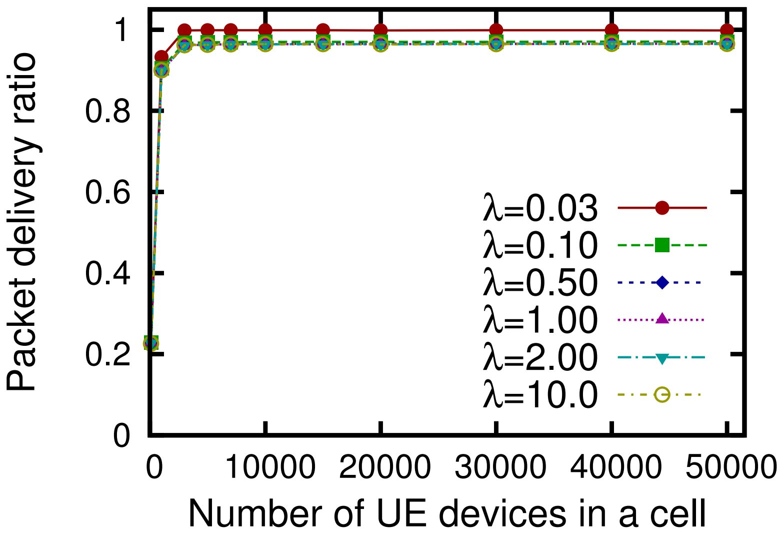
5.3.2 Discovery in Multi-cells

When an IoT device is located in a cell edge area, the following piratical association cases should be considered. If the IoT device is connected to a relay UE device which is connected to an eNodeB of an adjacent cell, the eNodeB can discover the relay UE device with the ProSe-based discovery, but they cannot communicate because the relay UE is connected to the adjacent eNodeB. Strictly speaking, multi-cell operation need be additionally designed to handle the above cases, but such operation is heavy. The author considers that a simple solution to make an IoT device in a cell edge area connect multiple relay UE devices reduces the possibility of the piratical association.

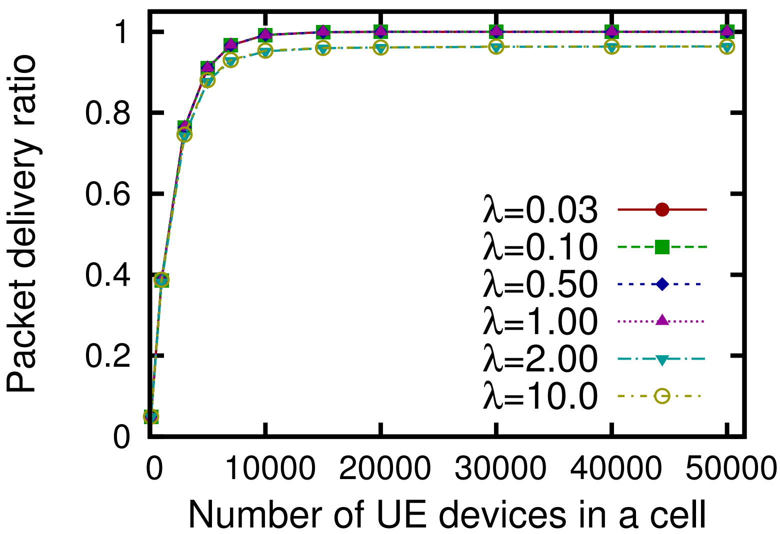
As for a state of relay UE device, relay UE devices are prevented from transiting to the RRC_IDLE state for discovery processes. Since an UE device becomes a relay UE device for a limited time, battery consumption as a relay UE device is not a serious issue.

5.4 Conclusion

This chapter has proposed new mobile architecture based on cellular networks to accommodate non-cellular fixed IoT devices by UE devices working as IoT gateways. IoT gateways, called relay UE devices transfer data between IoT devices and the Internet. The key features of the architecture are summarized below: First, its NDN based design avoids round trip delay which would be incurred if host-based communication were adopted. Second, the architecture leverages proximity services (ProSe) to discover relay UE devices with low overhead in terms of discovery messages.

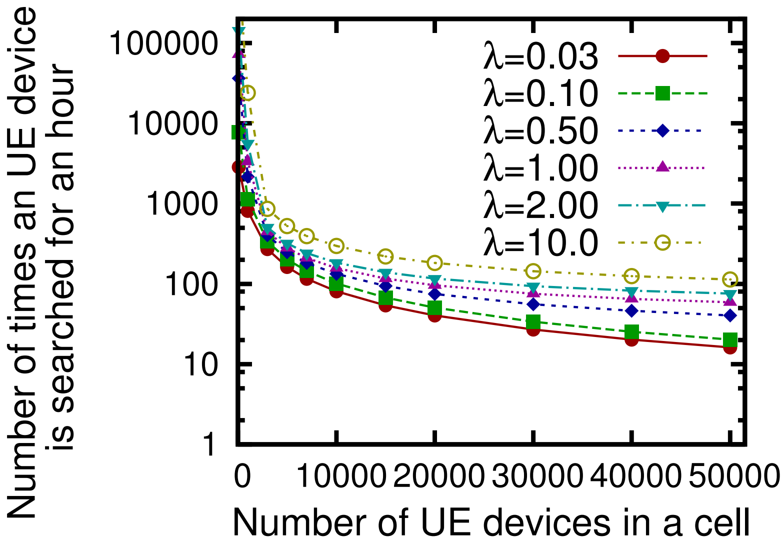


(a) Cell radius = 400 [m]

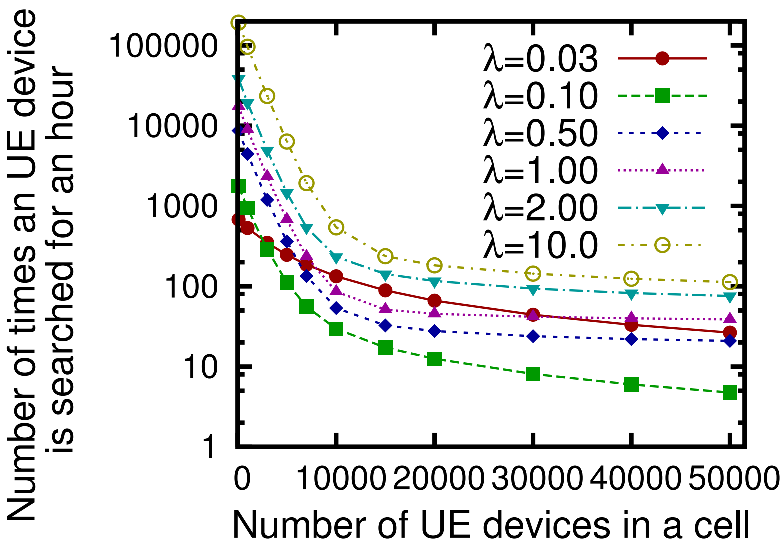


(b) Cell radius = 1,000 [m]

Figure 5.5: The relation between the number of UE devices in a cell and the packet delivery ratio (the number of fixed IoT devices: 100, mobility model of UE devices: truncated Lévy walk, interest packet arrival process: Poisson, interest packet arrival rate λ [packets/sec]: 0.03–10.0, the number of allowed relay UE connections k: 1)

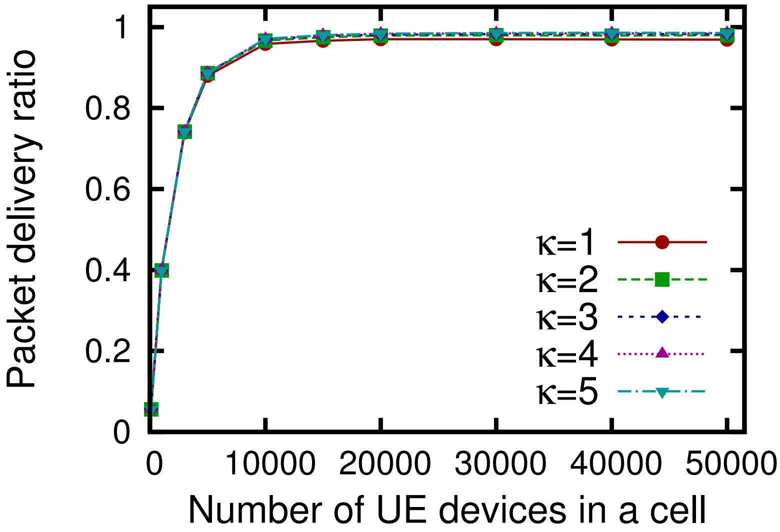


(a) Cell radius = 400 [m]

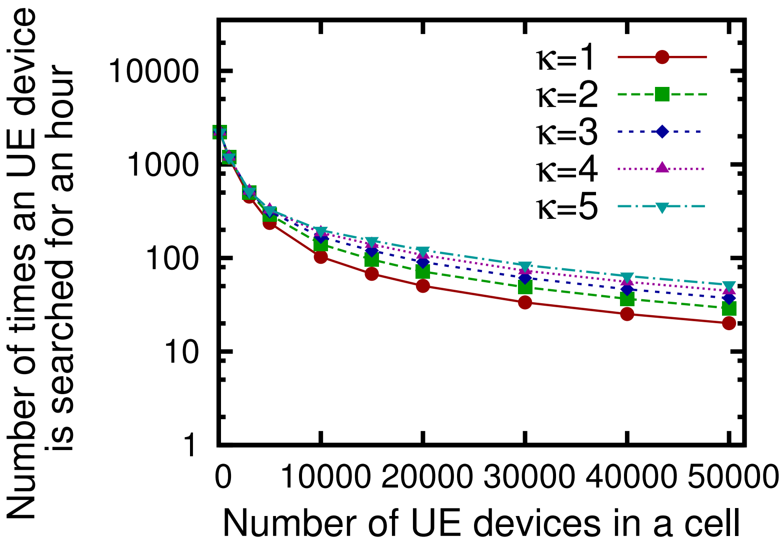


(b) Cell radius = 1,000 [m]

Figure 5.6: The relation between the number of UE devices in a cell and the number of times an UE device is searched for an hour. (the number of fixed IoT devices: 100, mobility model of UE devices: truncated Lévy walk, interest packet arrival process: Poisson, interest packet arrival rate λ [packets/sec]: 0.03–10.0, the number of allowed relay UE connections k: 1)



(a) Packet delivery ratio



(b) Discovery overhead

Figure 5.7: The number of UE devices in a cell versus the evaluation metrics when IoT devices are allowed to establish multiple relay UE connections. (cell radius [m]: 1,000, the number of fixed IoT devices: 100, mobility model of UE devices: truncated Lévy walk, interest packet arrival process: Poisson, interest packet arrival rate λ [packets/sec]: 0.10, the number of allowed relay UE connections k: 1–5)

Chapter 6

Conclusion

The thesis sheds light on mobility management in an IoT era because strict mobility management of current mobile networks such as cellular networks and mobile IP networks is richer than IoT communications require. For example, seamless handover with large communication and state overheads are not needed. The thesis re-designs mobility management assuming that core networks are cellular networks. Since many resource-constrained IoT devices, i.e., non-cellular IoT devices, do not have cellular interfaces, smartphone-based IoT gateway which connect non-cellular IoT devices to a cellular network is introduced. The thesis proposes the mobile architecture based on routing-based mobility management among cells by leveraging communication patterns of IoT devices. The mobile architecture comprises two schemes: the macro mobility management and the IoT gateway discovery. The macro mobility management scheme manages locations of cellular IoT devices like smartphones, whereas the IoT gateway discovery mechanism manages mapping between smartphone-based IoT gateways and non-cellular IoT devices.

First, the author has addressed how to propagate routing information of a huge number of mobile devices, which incurs a problem for macro mobility among cells, without extra overheads of holding and updating the routing information. In Chapter 4, the author has proposed a routing-based mobility management scheme using Bloom Filters to represent routing information for IoT devices by leveraging low mobility patterns of IoT devices. Through simulations in large-scale wireless mobile networks, the author has validated its feasibility in terms of overheads of routing information and the packet delivery rate degradation caused by periodical routing information updates. Specifically, the author has shown that the scheme can reduce the size of routing information, which are stored at routers, to approximately 73 Kbytes in the case that required packet delivery rate to target mobile devices is relaxed to around 0.9. Then the author applies the idea to OSPF, a scalable link-sate routing protocol, to validate scalability of routing-based mobility management.

Second, the author has addressed how to handle micro mobility of non-cellular IoT devices in an user-provided network based on a cellular network. In Chapter 5, the author has introduced a new relay mobile device discovery mechanism to accommodate non-cellular IoT devices in cellular networks by using smartphones as mobile IoT gateways. The discovery mechanism leverages long-range communication capabilities of LTE-based proximity services called ProSE for an edge router next to a base station to discover a relay UE device which is near by a target IoT device. A key technique is to search a target IoT device from former relay UE devices. The author has validated that the proposed discovery mechanism enable to deploy cellular-based user-provided networks which are feasible to accommodate a huge number of non-cellular IoT devices without extra financial costs and high discovery overheads.

Finally, the contributions of the thesis are summarized below:

1. The author re-designs mobility management suitable to IoT devices, whereas most IoT communication protocols use existing mobility management schemes provided by cellular and mobile IP networks.

2. The author proposes a new routing-based mobility management mechanism, which can achieve low overheads in terms of size of routing tables and route update messages, for low-mobility IoT devices.

3. To accommodate non-cellular IoT devices in cellular networks without wasteful resource consumptions of UE devices and base stations, I design and evaluate a new relay mobile device discovery mechanism by leveraging proximity services (ProSe).

Bibliography

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," *Future Generation Computer Systems*, vol. 29, pp. 1645–1660, Sept. 2013.

[2] K. Ashton, "That 'Internet of Things' thing," *RFiD Journal*, vol. 22, pp. 97–114, July 2009.

[3] D. Evans, "The internet of things: How the next evolution of the internet is changing everything," *Cisco White Paper*, Apr. 2011.

[4] 3GPP, "Service requirements for machine-type communications (MTC); Stage 1," TS 22.368, 3rd Generation Partnership Project (3GPP), Oct. 2014.

[5] T. Zachariah, N. Klugman, B. Campbell, J. Adkins, N. Jackson, and P. Dutta, "The Internet of Things has a gateway problem," in *Proceedings of International Workshop on Mobile Computing Systems and Applications*, HotMobile'15, pp. 27–32, ACM, Feb. 2015.

[6] M. T. Lazarescu, "Design of a WSN platform for long-term environmental monitoring for IoT applications," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, pp. 45–54, Mar. 2013.

[7] C. Perkins, "IP mobility support for IPv4, revised," RFC 5944, Internet Engineering Task Force (IETF), Nov. 2010.

[8] C. Perkins, "Mobility support in IPv6," RFC 6275, Internet Engineering Task Force (IETF), July 2011.

[9] M. Ishino, Y. Koizumi, and T. Hasegawa, "A study on a routing-based mobility management architecture for IoT devices," in *Proceedings of IEEE International Conference on Network Protocols*, ICNP '14, PhD Forum, pp. 498–500, IEEE, Oct. 2014.

[10] M. Ishino, Y. Koizumi, and T. Hasegawa, "A routing-based mobility management scheme for IoT devices in wireless mobile networks," *IEICE Transactions on Communications*, vol. E98.B, pp. 2376–2381, Dec. 2015.

[11] A. Broder and M. Mitzenmacher, "Network applications of bloom filters: A survey," *Internet Mathematics*, vol. 1, pp. 485–509, Jan. 2004.

[12] M. Yu, A. Fabrikant, and J. Rexford, "BUFFALO: Bloom filter forwarding architecture for large organizations," in *Proceedings of International Conference on Emerging Networking Experiments and Technologies*, CoNEXT '09, pp. 313–324, ACM, Dec. 2009.

[13] S. Tarkoma, C. E. Rothenberg, and E. Lagerspetz, "Theory and practice of bloom filters for distributed systems," *IEEE Communications Surveys and Tutorials*, vol. 14, pp. 131–155, Feb. 2012.

[14] Y. Wang, T. Pan, Z. Mi, H. Dai, X. Guo, T. Zhang, B. Liu, and Q. Dong, "Namefilter: Achieving fast name lookup with low memory cost via applying two-stage bloom filters," in *Proceedings of IEEE International Conference on Computer Communications*, pp. 95–99, Apr. 2013.

[15] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: Current state and future challenges," *IEEE Communications Magazine*, vol. 49, pp. 32–39, Nov. 2011.

[16] 3GPP, "Feasibility study for proximity services (ProSe) (Release 12)," TR 22.803, 3rd Generation Partnership Project (3GPP), Aug. 2012.

[17] 3GPP, "Study on architecture enhancements to support proximity-based services (ProSe) (Release 12)," TR 23.703, 3rd Generation Partnership Project (3GPP), Mar. 2014.

[18] 3GPP, "Proximity-based services (ProSe); Stage 2 (Release 13)," TS 23.303, 3rd Generation Partnership Project (3GPP), Dec. 2015.

[19] Z. ZhuPerkins, R. Wakikawa, and L. Zhang, "A survey of mobility support in the Internet," RFC 6301, Internet Engineering Task Force (IETF), July 2011.

[20] F. Pong and N.-F. Tzeng, "Concise lookup tables for IPv4 and IPv6 longest prefix matching in scalable routers," *IEEE/ACM Transactions on Networking*, vol. 20, pp. 729–741, June 2012.

[21] G. Chatzimilioudis, A. Konstantinidis, C. Laoudias, and D. Zeinalipour-Yazti, "Crowdsourcing with smartphones," *IEEE Internet Computing*, vol. 16, pp. 36–44, Sept. 2012.

[22] X. Sheng, J. Tang, X. Xiao, and G. Xue, "Sensing as a service: Challenges, solutions and future directions," *IEEE Sensors Journal*, vol. 13, pp. 3733–3741, Oct. 2013.

[23] H. Ma, D. Zhao, and P. Yuan, "Opportunities in mobile crowd sensing," *IEEE Communications Magazine*, vol. 52, pp. 29–35, Aug. 2014.

[24] W. Z. Khan, Y. Xiang, M. y Aalsalem, and Q. Arshad, "Mobile phone sensing systems: A survey," *IEEE Communications Surveys Tutorials*, vol. 15, pp. 402–427, Feb. 2013.

[25] F. Calabrese, L. Ferrari, and V. D. Blondel, "Urban sensing using mobile phone network data: A survey of research," *ACM Computing Surveys*, vol. 47, Nov. 2014.

[26] L. Zhang and et al., "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, pp. 66–73, July 2014.

[27] 3GPP, "General packet radio service (GPRS) enhancements for evolved universal terrestrial radio access network (E-UTRAN) access," TS 23.401, 3rd Generation Partnership Project (3GPP), Sept. 2014.

[28] X. Lin, J. G. Andrews, A. Ghosh, and R. Ratasuk, "An overview of 3GPP device-to-device proximity services," *IEEE Communications Magazine*, vol. 52, pp. 40–48, Apr. 2014.

[29] M. Ishino, Y. Koizumi, and T. Hasegawa, "Leveraging proximity services for relay device discovery in user-provided IoT networks," in *Proceedings of IEEE World Forum on Internet of Things*, WF-IoT'15, pp. 553–558, IEEE, Dec. 2015.

[30] X. Tie, A. Sharma, and A. Venkataramani, "A global name service for a highly mobile internet," *UMASS Computer Science Technical Report: UM-CS-2013-023*, pp. 1–19, 2013.

[31] D. Raychaudhuri, K. Nagaraja, and A. Venkataramani, "MobilityFirst: A robust and trustworthy mobility-centric architecture for the future internet," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 16, pp. 2–13, Dec. 2012.

[32] W. Luo and S. Tricci, "Distributed mobility management approaches with IPv6 prefix properties," *IETF Draftm, draft-luo-dmm-with-ipv6-prefix-properties-00*, Oct. 2012.

[33] D. Raychaudhuri, K. Nagaraja, and A. Venkataramani, "Mobilityfirst: A robust and trustworthy mobility-centric architecture for the future internet," *SIGMOBILE Mobile Computing and Communications Review*, vol. 16, pp. 2–13, Dec. 2012.

[34] F. Guist, L. Cominardi, and C. J. Bernardos, "Distributed mobility management for future 5G networks: Overview and analysis of existing approaches," *IEEE Communications Magazine*, vol. 53, pp. 142–149, Jan. 2015.

[35] C. J. Bernardos, A. D. L. Oliva, and F. Giust, "A PMIPv6-Based solution for distributed mobility management," *IETF Draft, draft-bernardos-dmm-pmip-06*, Mar. 2016.

[36] J. Augé, G. Carofiglio, G. Grassi, L. Muscariello, G. Pau, and X. Zeng, "Anchor-less producer mobility in icn," in *Proceedings of ACM Conference on Information-Centric Networking*, pp. 189–190, Sept. 2015.

[37] Y. Zhang, A. Afanasyev, J. Burke, and L. Zhang, "A survey of mobility support in named data networking," in *Proceedings of IEEE Workshop on Named-Oriented Mobility*, Apr. 2016.

[38] H. Ballani, Y. Chawathe, S. Ratnasamy, T. Roscoe, and S. Shenker, "Off by default!," in *Proceedings of ACM Workshop on Hot Topics in Networks*, Hotnets-IV, ACM, Nov. 2005.

[39] C. Perera, D. S. Talagala, C. H. Liu, and J. C. Estrella, "Energy-efficient location and activity-aware on-demand mobile distributed sensing platform for sensing as a service in IoT clouds," *IEEE Transactions on Computational Social Systems*, vol. 2, pp. 171–181, Dec. 2015.

[40] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong, "On the levy-walk nature of human mobility," *IEEE/ACM Transactions on Networking*, vol. 19, pp. 630–643, June 2011.

[41] V. Jacobson *et al.*, "Networking named content," in *Proceedings of International Conference on Emerging Networking Experiments and Technologies*, pp. 1–12, ACM, Dec. 2009.

[42] Wi-Fi Alliance, "Neighbor awareness networking technical specification v1.0," May 2015.

[43] Wi-Fi Alliance, "Wi-Fi Peer-to-Peer Services (P2Ps) technical specification v1.2," July 2015.

[44] Y. Nishiyama, M. Ishino, Y. Koizumi, T. Hasegawa, K. Sugiyama, and A. Tagami, "A proposal on routing-based mobility architecture for ICN-based cellular networks," in *Proceedings of IEEE Workshop on Name-Oriented Mobility*, NOM'16, pp. 780–785, IEEE, Apr. 2016.

[45] S. Mumtaz, K. M. S. Huq, and J. Rodriguez, "Direct mobile-to-mobile communication: Paradigm for 5G," *IEEE Wireless Communications*, vol. 21, pp. 14–23, Oct. 2014.

[46] G. Piro, A. Orsino, C. Campolo, G. Araniti, G. Boggia, and A. Molinaro, "D2D in LTE vehicular networking: System model and upper bound performance," in *Proceedings of International Congress on Ultra Modern Telecommunications and Control Systems*, ICUMT'15, pp. 281–286, IEEE, Oct. 2015.

[47] L. Lei, Z. Zhong, C. Lin, and X. Shen, "Operator controlled device-to-device communications in LTE-advanced networks," *IEEE Wireless Communications*, vol. 19, pp. 96–104, June 2012.

[48] E. Veloso, V. Almeida, W. Meira, A. Bestavros, and S. Jin, "A hierarchical characterization of a live streaming media workload," *IEEE/ACM Transactions on Networking*, vol. 14, pp. 133–146, Feb. 2006.