INVITED PAPER *Special Section on Wired-and-Wireless Network System Technologies in Beyond 5G/6G*

D2EcoSys: Decentralized Digital Twin EcoSystem Empower Co-Creation City-Level Digital Twins

Kenji KANAI[†,††a)], **Hidehiro KANEMITSU**[†††], **Taku YAMAZAKI**[††††], **Shintaro MORI**[†††††], ***Members***, **Aram MINE**[††††††], ***Nonmember***, **Sumiko MIYATA**[††††], ***Member***, **Hironobu IMAMURA**[*], ***Nonmember***, ***and*** **Hidenori NAKAZATO**[††], ***Fellow***

SUMMARY A city-level digital twin is a critical enabling technology to construct a smart city that helps improve citizens' living conditions and quality of life. Currently, research and development regarding the digital replica city are pursued worldwide. However, many research projects only focus on creating the 3D city model. A mechanism to involve key players, such as data providers, service providers, and application developers, is essential for constructing the digital replica city and producing various city applications. Based on this motivation, the authors of this paper are pursuing a research project, namely Decentralized Digital Twin EcoSystem (D2EcoSys), to create an ecosystem to advance (and self-grow) the digital replica city regarding time and space directions, city services, and values. This paper introduces an overview of the D2EcoSys project: vision, problem statement, and approach. In addition, the paper discusses the recent research results regarding networking technologies and demonstrates an early testbed built in the Kashiwa-no-ha smart city.

key words: *digital twin, smart city, p2p network, decentralized networking and computing, participatory sensing*

1. Introduction

A city-level digital twin is a critical enabling technology to construct a smart city that helps improve citizens' living conditions and quality of life. The article [1] introduces that a digital twin-based smart city will become a new starting point for smart cities' construction and produce enormous innovations to enable cities to be more innovative.

A digital twin is a technology that offers the creation of a very accurate digital model from the physical world. We can use simulation with the digital model to accurately understand the product's behavior and performance [2]. A seamless connection and real-time data exchange are essential to synchronize physical and virtual space: "twinning" of the digital to the physical [3].

The city-level digital twin aims to create a digital replica city [4]. The digital replica city includes geometry information of the city landscape, social infrastructure, and city activities. In addition, the city's information must be represented using standard information models, such as geographic information (CityGML) and the building information model (BMI). As described the city by the information model, the digital replica city can be used for various city applications, such as future city planning, traffic simulation, disaster simulation, and wireless network simulation.

Currently, research and development regarding the digital replica city are pursued worldwide. The first country-scale digital twin is known as Singapore's digital twin. The Singapore technology company, VIZZIO, creates a 3D city model of the entire Singapore using satellite photos and machine learning techniques [5]. In Japan, the Ministry of Land, Infrastructure, Transport and Tourism (MLIT) is also pursuing the PLATEAU project to create a country-scale 3D city model of Japan and publish the data as open data [6].

However, the projects only focus on creating the 3D city model and lack a mechanism to involve various key players, such as data providers, service providers, and application developers. The mechanism is essential for constructing the digital replica city and producing various city applications. In other words, creating an ecosystem is mandatory to advance (and self-grow) the digital replica city regarding time and space directions, city services, and values.

Based on this motivation, the authors of this paper are pursuing a research project, namely Decentralized Digital Twin EcoSystem (D2EcoSys) [7]–[9]. The project aims to realize the ecosystem of co-creation digital twins with city dwellers using promising network technologies, such as a millimeter wireless mesh network, decentralized networking, cloud-native technology, and blockchain. The project is supported by the National Institute of Information and Communications Technology (NICT)'s Beyond 5G&D Promotion Project, grant number 05601.

This paper introduces an overview of the D2EcoSys project: vision, problem statement, and approach. In addition, the paper discusses the recent research results regarding networking technologies and demonstrates an early testbed built in the Kashiwa-no-ha smart city.

The rest of the paper is organized as follows. Section 2 introduces the overview of D2EcoSys project, including the project's vision, problem statement, requirements, and approach. Sections 3, 4, and 5 demonstrate our recent research

Manuscript received May 9, 2023.
Manuscript revised August 16, 2023.
Manuscript publicized October 26, 2023.

[†]The author is with The University of Tokyo, Tokyo, 113-8654 Japan.
[††]The authors are with Waseda University, Tokyo, 169-8555 Japan.
[†††]The author is with Tokyo University of Technology, Hachioji-shi, 192-8548 Japan.
[††††]The authors are with Shibaura Institute of Technology, Saitama-shi, 337-8570 Japan.
[†††††]The author is with Fukuoka University, Fukuoka-shi, 814-0180 Japan.
[††††††]The author is with GaiaX Co., Ltd., Tokyo, 102-0093 Japan.
[*]The author is with HAFT Inc., Kashiwa-shi, 277-0814 Japan.
a) E-mail: kkanai@g.ecc.u-tokyo.ac.jp

DOI: 10.1587/transcom.2023WWI0001

efforts regarding the D2EcoSys project. Section 3 shows ICN-based service mesh for autonomous network and service creation. Section 4 represents a name-based efficient content discovery for IPFS to achieve low latency and flexible content search. Section 4 demonstrates three testbed environments of our project:

- ✓ A 60 GHz millimeter wireless mesh network testbed.
- ✓ An ICN-based wireless sensor network testbed.
- ✓ A self-standing smart pole for creating a wireless drone mesh network testbed.

Finally, Sect. 5 gives the conclusion and future work.

2. D2EcoSys Project

2.1 Our Vision

As described in the Introduction, the ultimate goal of the D2EcoSys project is to create the ecosystem of a co-creation digital twin city with dwellers at the Kashiwa-no-ha smart city. The mission of Kashiwa-no-ha smart city is to make a global vision a reality and involve critical players from diverse fields to shape and share a vision of the future with the world [10]. Inspired by the mission, the D2EcoSys project empowers such co-creation and mainly seeks how promising network technologies can contribute to co-creating a digital replica city.

Our idea is based on participatory sensing [11], [12], also known as cloud sourcing. As shown in Fig. 1, city dwellers could help collect the city's information on their own devices and provide the collected data for creating and updating digital twins. The city dwellers could be data providers and sell their collected data to city service providers.

Every city information, including 3D city model and "live" city activities, must be collected to build an up-to-date digital replica city. Road/street monitoring, air quality monitoring, and wireless signal map are straightforward examples of city activity monitoring. Recently, the latest 3D point cloud map is also important to realize self-driving car/robot services. A 3D point cloud map is used for localization and mapping. These services could be indispensable for the future of smart cities.

It takes high cost when service providers or city administrators are responsible for sensing all city activities. The required sensing devices and data processing services are diverse to recognize the activities. It may be possible to provide limited services, but the service availability and coverage expansion still need to be investigated.

Therefore, participatory sensing is a practical solution to co-creating a digital replica city. In addition, the mechanism to involve various key players, such as data providers, service providers, and application developers, is essential to self-grow the digital replica city in terms of both time and space directions and city services, and values.

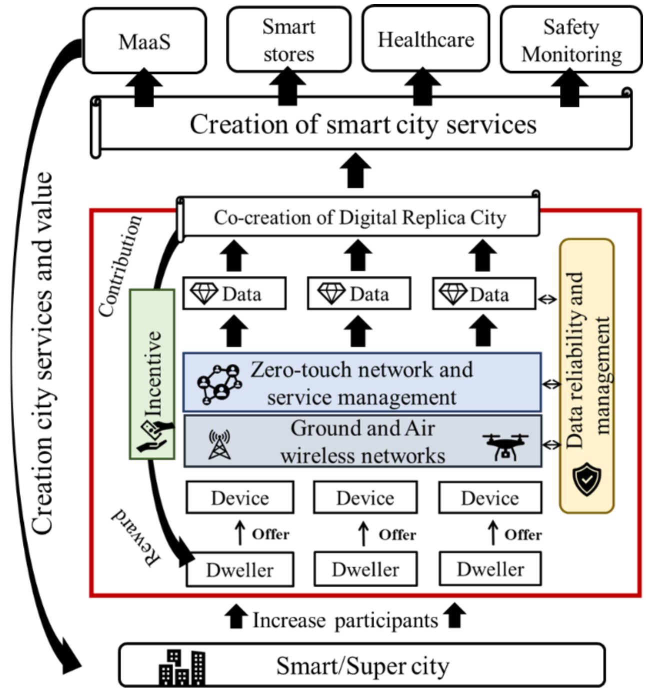


Fig. 1 Overview idea of an ecosystem of co-creation digital twins with city dwellers.

2.2 Problem Statement

Participatory sensing needs to be enhanced to realize our idea. The current participatory sensing contains the following issues:

1. **Centralized architecture:** All functionalities are concentrated in the cloud. Cloud has every responsibility, such as data storage, data processing, data accessibility, and data security. The centralized architecture invokes a single point of failure and load concentration. Scalability is limited.
2. **Incentive mechanism design:** There needs to be a concrete incentive mechanism to keep providing sensing data for dwellers. It requires an effective incentive mechanism to maintain the motivation of serving as a data provider.
3. **Data ownership:** The data ownership is delegated to service providers after providing the sensing data. Because the data can be managed and controlled by the service providers, they can limit the data accessibility or quit their services. This data governance leads to a barrier to data coordination and the creation of new services. In addition, it requires a privacy protection mechanism for dwellers.

Inspired by the former research efforts [13]–[15], the D2EcoSys project adopts a decentralized (or peer-to-peer based) participatory sensing to solve the above issues. Figure 2 shows an image describing the difference between centralized and decentralized participatory sensing. As shown

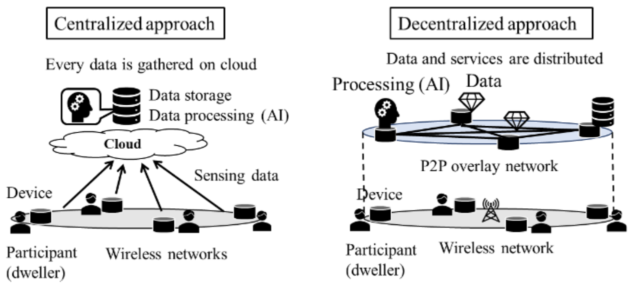


Fig. 2 An image describing the difference between centralized and decentralized participatory sensing.

in the figure, the decentralized participatory sensing keeps storing the data (and copying the data if necessary) in the participant devices. The data is distrusted via a Peer-to-Peer (P2P) overlay network. We assume that distributed computing is also supported over the P2P networks. In other words, the participant devices provide processing capability as well as sensing capability. The decentralized system design could be practical to realize the project's vision, as illustrated in Sect. 2.1.

2.3 Basic Requirements

In Sect. 2.3, we summarize the basic requirements of the co-creation of a digital twin based on decentralized participatory sensing as follows:

1. **Usability:** Zero-touch operation must be required because participants (or dwellers) are only sometimes familiar with ICT technologies. Once the participant registers the participatory sensing, all processes must be reliably and autonomously operated without complex configurations. Example processes are P2P network establishment, data and service discoveries, data and service distributions, data retrieval, and service invocation.
2. **Availability:** Effective incentive mechanisms must be required not to keep the participants but also to grow the participation (e.g., increase the number of participants, device types, and service types). Data and service reachability must be ensured. In the case of wireless P2P networks, the network topology might be changed dynamically because the peer nodes could quickly enter or break off the network (e.g., high churn rate) regardless of the participants' intentions. This situation degrades the system's availability.
3. **Reliability:** Data screening and data integrity must be required not to malfunction services and applications. Because the participants are only sometimes data scientists, they might perform a data poisoning attack (injecting the junk and poisoned data into the system) regardless of their intentions.

2.4 Our Approach

The D2EcoSys project set three work packages (WPs) to establish underlying technologies to meet the above requirements.

WP 1: Zero touch network and service management
This work package aims to establish a zero-touch network and service management technology to satisfy the requirements of usability and availability as described in Sect. 2.3. In WP 1, there are three tasks as follows:

- ✓ **Zero touch network and service invocation**: This task is to design and implement a zero-touch network and service invocation method using Information-Centric Networking (ICN) [16], [17] and cloud-native technology [18], such as container virtualization and container orchestration. For the early prototype, we proposed ICN-based Service Mesh [19] and introduce the proposed method in Sect. 3.
- ✓ **Efficient and flexible data and service discovery**: This task is to develop an efficient and flexible data and service discovery method for InterPlanetary File System (IPFS), a well-known P2P-based file-sharing system. We proposed KadRTT [20], [21] and successfully suppress the data discovery time compared to the original Kademlia [22] adopted in IPFS. KadRTT is introduced in Sect. 4.
- ✓ **Gamification-based incentive mechanism:** This task is to design the incentive mechanism using gamification, similar to the related work [24]. Contribution estimation and value distribution models are under development. These models will be implemented as a smart contract of blockchain. The very early gamification-based incentive model is proposed in [25]. The proposal uses a coalitional game to model a co-creative data transaction game focusing on the data consumer side.

WP 2: Data reliability and availability management
This work package aims to establish data reliability and availability management technologies to satisfy the reliability and availability requirements described in Sect. 2.3. In WP 2, there are three tasks as follows:

- ✓ **Data quality estimation and risk evaluation:** This task is to develop methods for estimating data quality and evaluating potential risk for judging whether input data is acceptable or not (e.g., junk or poisoned). Data screening is an essential pre-processing in machine-learning and deep-learning (AI) tasks to maintain model accuracy. If junk and poisoned data are contained in the AI training phase, the AI might not work as expected. The developing method in this task could contribute to ensuring the safety of AI usage.
- ✓ **Data falsification detection system using blockchain:** This task is to design and implement a data falsification detection system using blockchain. In addition to detecting junk and poisoned data, data falsification detection is essential to improve data reliability. The developing system adopts blockchain to ensure data immutability (or completeness) and to detect data falsification [26]. The system can handle both sensing data

and the AI model.

- ✓ **Efficient data cache in wireless sensor networks:** This task is to develop a mechanism that enables efficient data cache for ensuring data and service reachability in wireless P2P (or sensor) networks. The early prototype is developed as a wireless sensor node with an ICN-based content cache mechanism and several wireless interfaces, such as Wi-Fi, Local 5G, and IEEE 802.11ay-based millimeter communication [27]. The prototype is introduced in Sect. 5.

WP 3: Implementation and demonstration
This work package aims to implement and demonstrate the proposed prototype ecosystem at Kashiwa-no-ha smart city. In WP 3, there are three tasks as follows:

- ✓ **System architecture and prototype**: This task is to design a system architecture of the proposed ecosystem that integrates all underlying technologies described in WPs 1 and 2. In addition, this task is to implement the prototype system. We develop an intelligent 3D sensor system as the prototype system using a LiDAR sensor. We assume that the 3D sensor system is critical for creating a digital replica city.
- ✓ **Testbed:** This task is to create a testbed environment to demonstrate and validate the prototype. The testbed environment is being built at KOIL MOBILITY FIELD [28], a development and experiment field of robots in Kashiwa-no-ha smart city. In KOIL MOBILIY FIELD, we will install wireless sensing nodes and self-standing smart poles to establish ground and air wireless mesh network environments. The early-developed testbed is demonstrated in Sect. 5.
- ✓ **Demonstration:** This task is to demonstrate the prototype of the ecosystem on the testbed and evaluate the performance of the ecosystem, including co-creation capability.

3. ICN-Based Service Mesh

This section introduces the proposed ICN-based service mesh. The content is an edited version of our paper [19]. The entire content is found in the original paper [19].

3.1 Objective

The ICN-based service mesh offers a name-based service invocation mechanism for autonomous network and service creation. Although a cloud-native approach and microservice architecture can create an ultra-flexible system and application, system development and management have become increasingly complicated. Autonomous management is an indispensable capability to reduce the burdens of developers and operators. Container-based network function virtualization and orchestration is a promising approach to realizing autonomy.

It is mandatory to shift from connectivity-based networking to "service/information"-centric networking. Native container-based network functions offer only host-based network connectivity among container-based functions in terms of networking. The host-based connectivity needs mechanisms, such as controller and function discovery, for autonomous networking and management. The centralized architecture has a scalable issue to support dynamic in-network computing and network-wide distributed computing. Controller and function discovery capabilities should shit from the application layer into the network layer (i.e., in-networking approach).

Based on this motivation, we proposed and developed the ICN-based service mesh. By inspiring the former research efforts regarding ICN-based Service Function Chaining (SFC) [29]–[34], we design and implement the ICN-SFC capability as a middleware and offer the "information-centric service mesh" to all containerized functions. We implement the service mesh using Cefore and Cefpyco [35], one of the CCNx software implementations developed by NICT in Japan.

We design the information-centric service mesh using an ambassador container style [36]. The ambassador container style brings us to decoupling sub-functionalities, such as networking capability, from the service functions. The sub-functionalities are implemented in the ambassador container and proxy a network connection instead of the service function. Based on the implementation design, we newly designed a function chain handling container (Chain Handler) to proxy the capability of ICN-SFC. The detailed explanations are referred to in the original paper [19].

3.2 Results

We demonstrated the performance of the information-centric service mesh. Figure 3 summarizes the service function chain models we adopt. We create a K8s cluster in our laboratory. The K8s cluster consists of one master node and eight worker nodes. Every data producer node publishes small images (approximately 15 KB), and all container-based functions transfer the input data (i.e., relay capability). We carry out 25 trials for each model.

Figure 4 shows the result of the average end-to-end

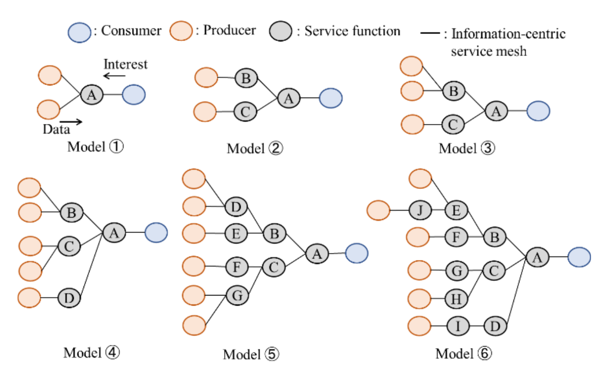


Fig. 3 Test service function chain models [19].

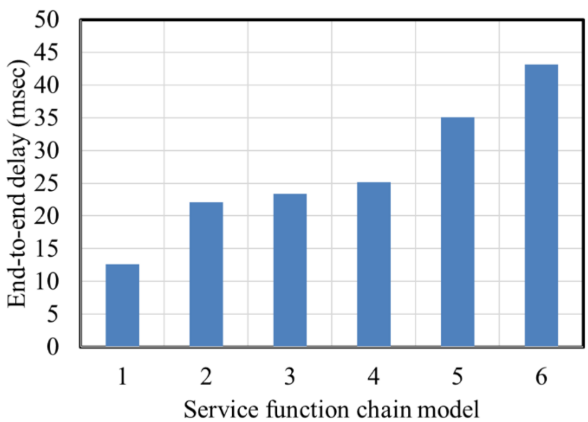


Fig. 4 End-to-end delay results of ICN-based service mesh [19].

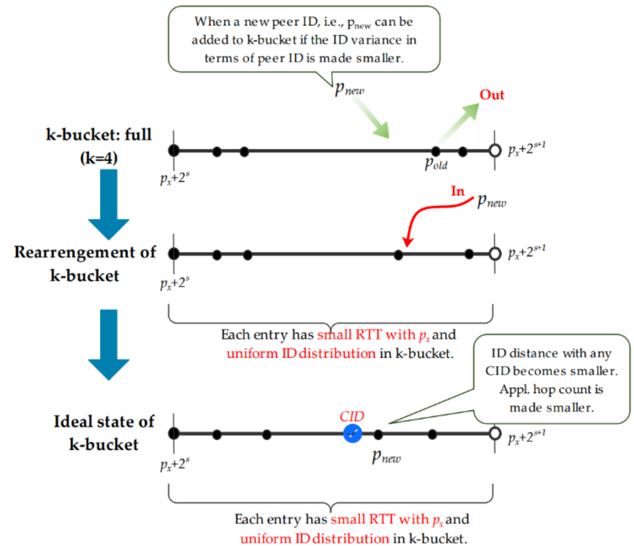


Fig. 5 ID rearrangement in KadRTT.

delay for six different chain models. From the results, the end-to-end delay increases as the number of hops of the chain increase, while the end-to-end delay changes little as the number of hops is the same. These results are reasonable and conclude that the information-centric service mesh works appropriately.

In the future, we will adopt more realistic functions regarding digital twin processing, such as point cloud streaming and processing, to validate the performance of the proposed service mesh. In addition, we integrate an efficient data and service discovery method, such as introduced in the next section, for efficient network and service management.

4. Name-Based Efficient Content Discovery for IPFS

This section introduces name-based efficient content discovery for IPFS. The content is an edited version of our paper [20], [21]. The entire content is found in the original paper [20], [21].

4.1 Objective

InterPlanetary File System (IPFS) [37], which has been applied to distributed file sharing and blockchain applications, is a critical middleware for Web3.

In the sensor network over the P2P network, each sensing device can interact with each other to autonomously obtain the input data for processing IoT applications. However, there are several challenging issues: (i) each required data must be retrieved with low latency, and (ii) a flexible search query method, such as range and multi-attribute queries, must be implemented for practical use. Our method for resolving both (i) and (ii) assume to use IPFS. The next subsection presents a routing algorithm named “KadRTT” for resolving (i).

4.2 Query Target Selection

KadRTT [20] has three features: (i) RTT-based lookup guaranteeing the maximum overlay hop count does not exceed that of the original Kademlia, (ii) the uniform ID arrangement for each k-bucket to shorten the initial ID distance to suppress the application hop count, and (iii) dynamic lookup parameter adjustment for minimizing the lookup latency. As for (i), CID be the target content ID, let p_i^{kad} be the peer ID in the k-bucket for Kademlia, and let p_i^{RTT} be the peer ID in k-bucket for KadRTT, respectively.

If we define the ID distance between x and y as $d(x, y)$, the following relationship holds for KadRTT.

$$\log\left(d\left(p_i^{RTT}, CID\right)\right) - \log\left(d\left(p_i^{kad}, CID\right)\right) < 1 \quad (1)$$
$$\leftrightarrow \frac{d(p_i^{RTT}, CID)}{d(p_i^{kad}, CID)} < 2$$

where we assume that p_i^{RTT} has a smaller RTT with the client than p_i^{kad}, but the ID distance between p_i^{RTT} and CID is larger than the one between p_i^{kad} and CID, because Kademlia attempts to select the nearest peer ID from CID in the k-bucket for each content lookup. However, if Eq. (1) holds, this means that selecting p_i^{RTT} in KadRTT as the next lookup target leads to equalize the upper bound of the application hop count as Kademlia. As a result, the lookup latency in KadRTT is made smaller by the difference of RTT for each lookup.

4.3 Uniform ID Arrangement in Routing Table

The second characteristic of KadRTT, i.e., (ii), is shown in Fig. 5. The variance in terms of the peer ID for each bucket is reduced by swapping an existing peer ID (p_{old}) with a new peer ID (p_{new}) during the lookup procedures. During lookup procedures, if the ID variance is reduced by swapping p_{old} with p_{new} and $RTT(p_x, p_{new}) < RTT(p_x, p_{old})$, p_{old} is discarded and p_{new} is added to the k-bucket, where p_x is a client peer and $RTT(x, y)$ is the round trip time from peer x to y. As a result, it is expected that the initial ID distance can be shorter than that of original Kademlia.

4.4 Lookup Parameter Optimization

Since a swapping procedure is triggered when the bucket is full, the time required to reach the state depends on each k-bucket size, i.e., k, and the lookup query arrival rate. Thus, we consider that the time duration until the k-bucket is full affects to the lookup latency. The lookup concurrency, i.e., α, affects the total lookup latency as well as the number of sent queries. Since a client peer sends α lookup queries simultaneously, the maximum lookup iteration count per bucket is $\lceil \frac{k}{\alpha} \rceil$. If α is smaller, the maximum iteration count increases, i.e., the worst lookup latency can be larger. By contrast, if α is larger, the worst lookup latency can be smaller, but more lookup messages must be exchanged. Consequently, the available bandwidth for each peer can be degraded. Optimizing α leads to the improvement of the lookup latency, and the number of next hops (next lookup targets returned from each queried peer), i.e., β, presents similar characteristics as α. Thus, k, α, and β must be appropriately adjusted according to the network conditions such as the iteration count, and the message arrival rate.

Firstly, we present the optimal value for k, i.e., the number of k-bucket. It is derived from the upper bound for the lookup latency and let the optimal value for k as k_{opt}. Subsequently, the optimal values for both α and β are derived using k_{opt}.

Figure 6 shows an k-bucket state at t in KadRTT. In this figure, the nearest ID distance from CID at t is bounded by the expected lookup latency at time t.

If we define $\overline{T_{LT}^{RTT}}$ at t as $\overline{T_{LT}^{RTT}}(t)$, it is defined as follows:

$$\overline{T_{LT}^{RTT}}(t) \le \frac{k\rho t_{ave}^{kad}}{\alpha} \log\left(\frac{\lambda^k e^{-\lambda} 2^r}{k}\right) = T_{kadRTT}^{t\to\infty} \qquad (2)$$

In particular, since it is necessary that only one iteration is required to minimize the lookup latency, the lookup

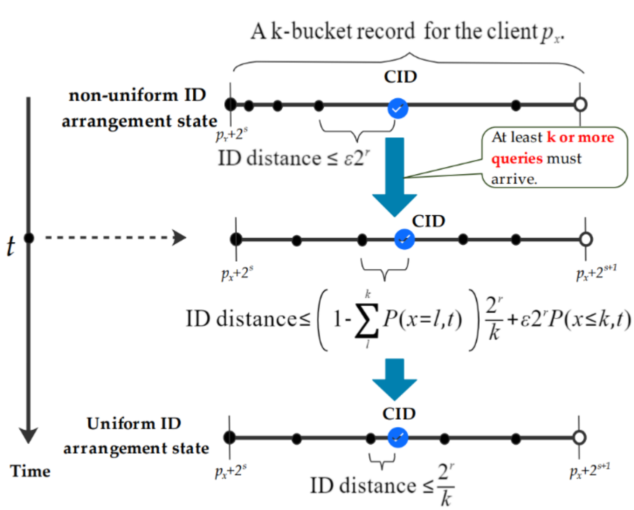


Fig. 6 k-bucket state at (k = 4).

latency per iteration $T_{kadRTT}^{t\to\infty}(r,1)$ is as follows:

$$T_{kadRTT}^{t\to\infty}(r,1) = \rho t_{ave}^{kad} \log\left(\frac{\lambda^k e^{-\lambda} 2^r}{k}\right) \qquad (3)$$

In Eq. (3), the solution to k when $T_{kadRTT}^{t\to\infty}(r,1)$ takes positive and minimum values as

$$k_{opt}(r) = \left\lceil -\frac{W\left(-2^r e^{-\lambda}\log_e \lambda\right)}{\log_e \lambda} \right\rceil \qquad (4)$$

In this section, we derive the optimal lookup concurrency, α, and the number of next hops (next candidates), β, using the expected number of iterations. In the previous section, we set the number of iterations to one to derive $k_{opt}(r)$ in Eq. (4) as the ideal condition. However, Eq. (3) does not contain either α or β. Thus, the expected number of iterations using both α and β is necessary to derive their optimal values. If we define the expected number of iterations as $E_{ite}(\alpha,\beta,k)$ can then be rewritten as

$$T_{kadRTT}^{t\to\infty}(r) = \rho t_{ave}^{kad} \mathrm{E}_{ite}(\alpha,\beta,k) \log\left(\frac{\lambda^k e^{-\lambda} 2^r}{k}\right) \qquad (5)$$

We then obtain $\alpha(r)$ and $\beta(r)$ when the number of lookup iteration is within only one, as follows:

$$\alpha(r) = \left\lceil |PL| \sqrt{\left| \frac{2k_{opt}(r)\log_e\left(1-P_{query}(r)\right)}{W\left(\frac{2}{k_{opt}(r)}|PL|^2 \log_e\left(1-P_{query}(r)\right)\right)} \right|} \right\rceil,$$
$$\beta(r) = \left\lceil \sqrt{\left| \frac{W\left(\frac{2}{k_{opt}(r)}|PL|^2 \log_e\left(1-P_{query}(r)\right)\right)}{2k_{opt}(r)\log_e\left(1-P_{query}(r)\right)} \right|} \right\rceil \qquad (6)$$

where $|PL|$ is the pool size, that is maintained during the lookup procedures for each peer, and function W is the *Lambert W function*. The pool contains the set of peer IDs returned from queried peers, i.e., the set of next hops.

Figure 7 shows how each peer updates those parameters. First, a peer attempts to update $k_{opt}(r)$, $\alpha(r)$, and $\beta(r)$ for every k-bucket index r. For each k-bucket index r, in Eq. (4), we derive $k_{opt}(r)$ using the current lookup message arrival rate in a predefined time interval. Although the time interval should be determined according to the assumed network load and scale, this is beyond the scope of this study. Subsequently, both $\alpha(r)$, and $\beta(r)$ are updated using $k_{opt}(r)$.

4.5 Comparison in Real Implementation

We conducted comparisons by testground [38], that can emulate peer-to-peer networks using containers as well as processes. We implemented both KadRTT and KadRTT2 using the libp2p implementation [39], then we containerized each peer on Amazon EKS (Elastic Kubernetes Service). In total we run 300–1200 containers, i.e., peers on EKS on 30 VMs with "C5.2xLarge" instance type. The comparison

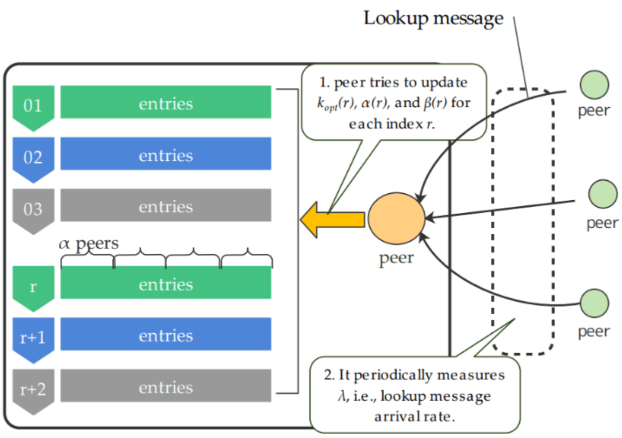


Fig. 7 Parameter calculation in KadRTT2.

Table 1 Execution parameters in testground.

Parameter	Value
# of peers (containers)	300, 600, 1200
Runner	cluster:k8s
Test case	“find-providers”
Network latency (ms)	100
k-bucket size (k) for kad-dht/KadRTT	20
Lookup concurrency (α) for kad-dht	10
# of next hops (β) for kad-dht	10
Content hit probability for KadRTT2	0.98
Measurement interval for KadRTT2 (s)	50

targets are “kad-dht”, which is a libp2p implementation for Kademlia. In particular, we implemented both KadRTT and KadRTT2 as libp2p implementations, which is available at [40]. The comparison metrics are the mean lookup latency, the content hit probability, and the number of lookup queries, that are obtained from log files in testground.

Table 1 shows the setup parameter in testground. As a preliminary experiment, we set the number of peers (dockerized containers) to be up to 1200, and $(k, \alpha, \beta) = (20, 10, 10)$, that is the same as the actual IPFS default values for kad-dht and KadRTT.

Table 2 shows the comparison results, where KadRTT2 outperforms others in terms of the lookup latency. The content hit probability among KadRTT and KadRTT2 is almost the same, and the number of lookup queries of KadRTT2 is slightly larger than that of KadRTT. From these results, KadRTT2, i.e., parameter optimization can contribute to reducing the lookup latency, while there is room for reducing the number of lookup queries.

The lookup parameter optimization as KadRTT2 presents an improved performance than KadRTT in real implementation in this limited network scale. The reason that the increase of exchanged messages in KadRTT2 is due to both α and β derived by our proposal. In the future, we investigate how such increases affect the actual content downloading performance.

Table 2 Comparison results on testground.

	FindFirst (avg.)		
# of peers	300	600	1200
kad-dht	2824.361	3287.5562	4411.6519
KadRTT	2345.854	2594.5145	3422.601
KadRTT2	1819.162	1993.8016	2512.2627

	Hit Rate		
# of peers	300	600	1200
kad-dht	73.037	85.341	82.123
KadRTT	99.684	99.443	99.567
KadRTT2	99.493	99.667	99.384

	# of queries		
# of peers	300	600	1200
kad-dht	28288	39943	52163
KadRTT	20345	28239	36845
KadRTT2	21146	29055	39141

5. Testbed Environment

This section introduces our testbed environment for the project. We are developing the testbed in KOIL MOBILITY FIELD (KMF) [28] at Kashiwa-no-ha smart city. KMF is known as a test field for robot development and performance validations. The field has a circuit course for testing self-driving micro mobilities. In addition, the field is available for wireless communications, such as Wi-Fi, 5G/Local 5G, and even 60 GHz millimeter wireless mesh networks. The 60GHz wireless mesh network is based on IEEE 802.11ay and standardized by Meta, called “Terragraph [41].” A detailed explanation of the Terragraph is referred to [41]. According to [41], Terragraph is available on a wireless Gbps network, and 60 GHz wave reaches near to 1km if the sky is clear and there are no obstacles (Line of Sight (LOS) environment).

Terragraph can establish private Gbps-class wireless “mesh” networks. Terragraph can quickly expand wireless network coverage to add an Access Point (AP) similar to Wi-Fi. This capability could be beneficial to support a network infrastructure for co-creating a city-level digital twin. A Gbps-class mesh network could help streaming 3D city models and 3D point clouds. The data size of the 3D model usually has over GB.

Based on this motivation, we are developing the testbed in KMF. An overview image of the testbed is shown in Fig. 8. As shown in the figure, we will install wireless sensing nodes and self-standing smart poles to establish ground and air wireless mesh network environments. In addition, the testbed is connected to the Internet, so we can remotely access the testbed, and the testbed can connect with clouds and other experiment fields.

5.1 60 GHz Millimeter Wireless Mesh Network

This subsection demonstrates the performance of a 60GHz

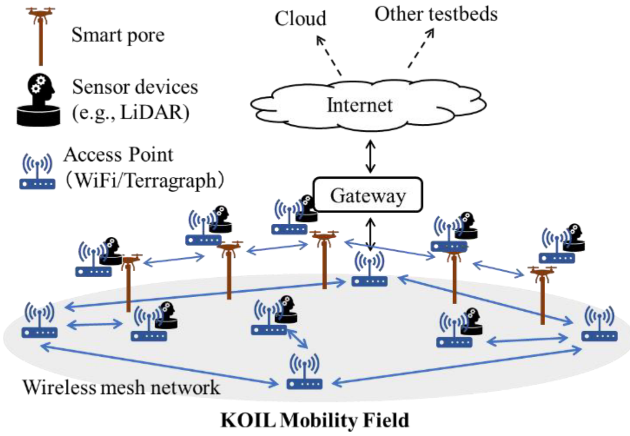


Fig. 8 An image of our testbed.

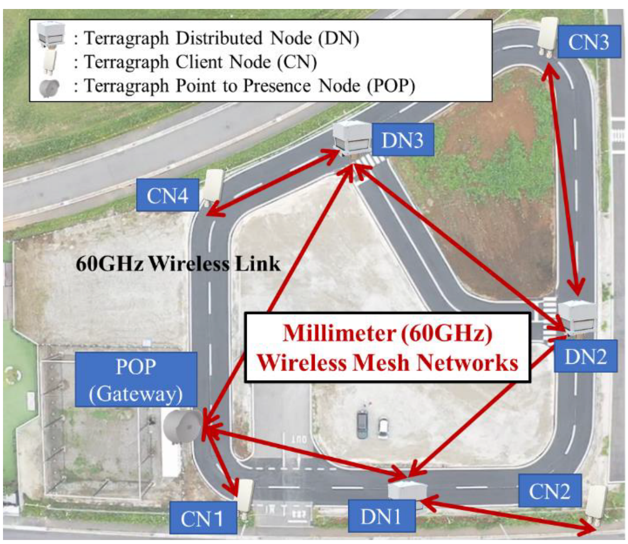


Fig. 9 Network topology of 60GHz millimeter wireless mesh network in KOIL MOBILITY FIELD.

millimeter wireless mesh network. As described in the previous section, the 60 GHz wireless mesh network has already been deployed in KMF and opened as mmWave Lab@Kashiwa-no-ha [42]. The network topology is shown in Fig. 9. The wireless mesh network consists of three types of AP of Terragraph: Distributed Node (DN), Client Node (CN), and Point to Presence (POP) node. DN can establish a Terragprah's mesh network, and CN is an endpoint of the mesh network and can connect to the other devices via ethernet. POP serves as a gateway of the mesh network.

We newly installed seven Mini PCs as network nodes in KMF, and each Mini PC is connected to each DN or CN via ethernet. A snapshot of the installation of the Mini PC is shown in Fig. 10, and the location of installations of Mini PCs is listed in Table 3. We implemented a network monitoring tool using iPerf3, PyPing, and Grafana. The monitoring tool is installed on the Mini PCs and has the capability of transmitting CUBIC-TCP flows and ICMP packets to each Mini PC for measurement of network bandwidth and latency.

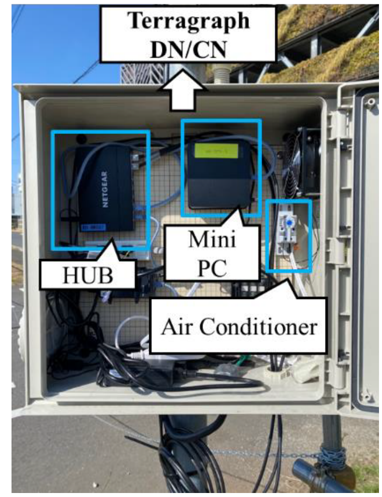


Fig. 10 A snapshot of installation of Mini PC.

Table 3 Location of installation of Mini PC.

Terragraph ID	Mini PC ID
CN1	KMF-1
DN1	KMF-2
CN2	KMF-3
DN2	KMF-4
CN3	KMF-5
DN3	KMF-6
CN4	KMF-7

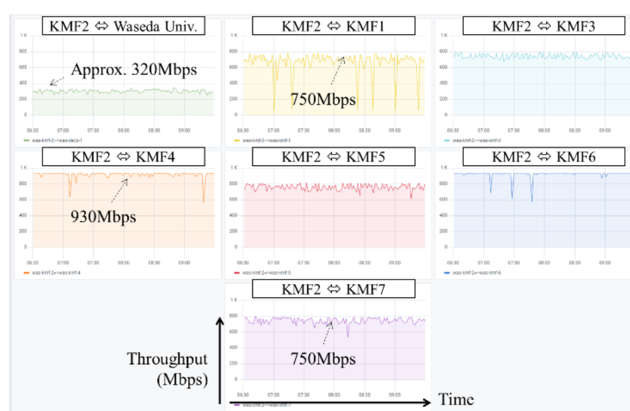


Fig. 11 A snapshot of the Grafana dashboard of our monitoring system.

We have been monitoring the performance of the wireless mesh networks since the end of March 2023. The performance is almost stable during the monitoring, even if heavy rainy days. The monitoring results show that the network bandwidth between DNs is approximately 930 Mbps, and between CN and DN is approximately 750 Mbps. The latency is around 3 msec regardless of the wireless links. A snapshot of the Grafana dashboard of our monitoring system is shown in Fig. 11.

5.2 ICN-Based Wireless Sensor Network Testbed

This subsection introduces the prototype ICN-based wireless sensor network testbed. Although we have a plan for the prototype installed in KMF, this subsection only explains an early-stage prototype and shows the unit testing results. The content is an edited version of our paper [27]. The entire content is found in the original paper [27].

In typical smart-city frameworks, sensor nodes (SNs) are directly connected to cloud servers to centralize sensing data. These frameworks have a structural issue, such as address-based networks on standard protocols creating a potential bottleneck for handling dynamic wireless data transmissions. In particular, when data retrieval, heavy address-based queries cause serious protocol overheads, and wireless forwarding may lead to extra network congestion and frame collisions. An ICN is a promising content-oriented network model to address the above issues as a similar motivation discussed in Sect. 3. ICN focuses on content rather than connectivity, i.e., it natively supports naming-based routing and in-network caching. The end-users in WSN systems are also interested in the data regardless of its location, so it makes sense to combine ICN with WSNs in what are known as information-centric wireless sensor networks (ICWSNs) [42].

ICWSNs can boost the effectiveness of data collection, delivery, and discovery thanks to regional caching and the cooperation of nodes. Specifically, any node can copy and store the caching data without receiving permission from the original publisher and reply to further requests without notifying the subscriber that the data has been provided from the cache memories. These features lead to a framework that is not only resource-efficient but also energy-efficient. Namely, the in-network caching scheme can efficiently handle the sensing data from unavailable nodes and minimize retrieval delay. A naming scheme can enable better data information management and an easy data-retrieval mechanism.

In our study [26], a testbed device was implemented for the on-site deployment of an ICWSN framework that could overcome the issues regarding long-term operation scenarios and outdoor environments. Figure 12 shows the prototype portable-unit-type testbed device that consists of Advantech AIR-020X (Six-core ARM v8.2 CPU, 8-GB RAM, Ubuntu 18.04 with Jet Pack OS) computer and peripheral devices. On the other hand, as a node supporting integrated ground-to-air ICWSNs, as shown in Fig. 12, the waterproof and anti-shock testbed device includes a single-board computer (Two-core 1.8GHz Intel Atom CPU, Ubuntu 20.04, and Intel AI and 4G/LTE cellular modules). In addition, it has a power supply from a three-cell lithium-ion battery, such as a power source to drive (aerial) vehicles.

Using these devices, for preliminary evaluation of the ICWSN system, we constructed a prototype network, as shown in Fig. 13. As shown in Fig. 14, we also conducted an experiment using Terragraph as the air interface for ICWSNs.

(a) (b)

Fig. 12 Overview of testbed device. (a) A portable device for evaluation as a testing sensor node. (b) A compact device available for ground and sky as a relay and mobile station node.

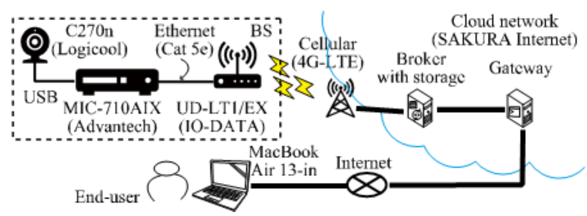


Fig. 13 Network model of ICWSN framework.

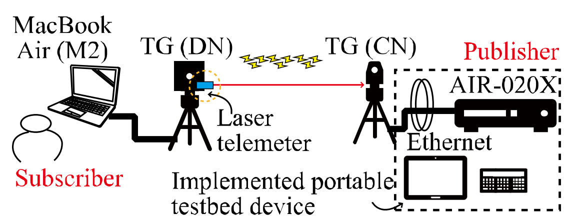
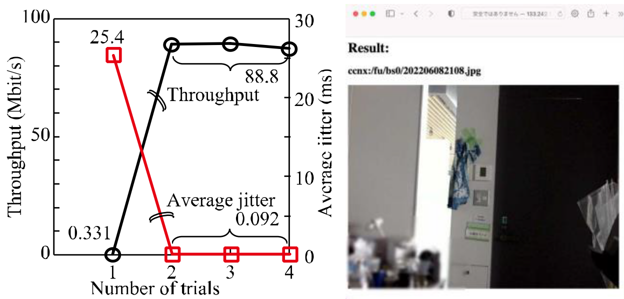


Fig. 14 Experimental setting of mmWave communications (TG: Terragraph).

Note that Terragraph is an IEEE 802.11 ad/ay-compliant platform developed by Meta (Facebook) and has been successfully deployed by mobile operators and Internet service providers for the backhaul in wireless mesh networks. The experiment includes throughput at the TCP layer using iPerf3 and the ICN layer using Cefore. Figure 15 shows the average throughput and jitter for requesting 20 (different) data, where each data comprises five trials of retrieval. The results here indicate that the end-user can stably and efficiently obtain data that are a low-delay and high-throughput. Compared to the conventional frameworks, the proposed scheme can reliably obtain the data anytime and anywhere due to its pull-type retrieval mechanism and distributed caching scheme, as shown in Fig. 15. On the other hand, as shown in Fig. 16, the results indicate that there was no significant throughput degradation depending on the distance between TG nodes, unlike that seen with IEEE 802.11 standards in Sub-6-GHz bands.



(a) (b)

Fig. 15 Implementation and evaluation results. (a) End-to-end network throughput and jitter vs number of requests to retrieve for same data. (b) A display screen of a data retrieving result based on a HTTP-based request sent from the end-user to the broker.

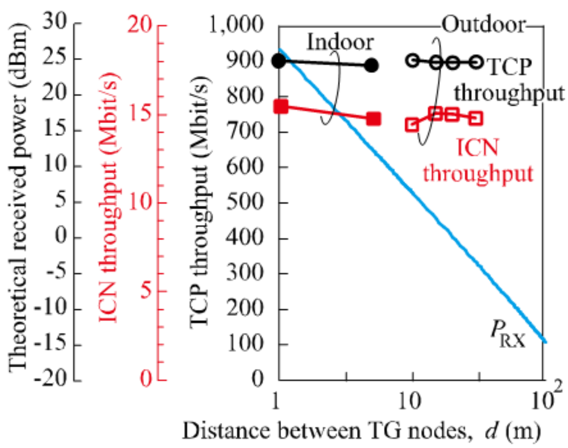


Fig. 16 TCP and ICN throughput and theoretical received power vs. distance between Terragraph nodes. (TG: Terragraph).

5.3 A Self-Standing Smart Pole for Creating Wireless Drone Mesh Network Testbed

This subsection introduces a self-standing smart pole to offer a wireless drone mesh network testbed. In addition, the subsection demonstrates the performance of actual air and ground wireless networks using the self-standing pole. The content is an edited version of our paper [43]. The entire content is found in the original paper [43].

To realize future smart cities, various services, such as delivery and monitoring tasks, should be provided using drones. D2EcoSys is expected to empower smart cities toward the safe and reliable provision of airspace. However, flying drones have various limitations while maintaining a drone network and communicating with each other owing to legal restrictions and ensuring safety for prototyping in real fields.

In this project, as a prototype of D2EcoSys, we have been developing a sensing system and its experiment environment based on the city airspace testbed (CAT) [43]. CAT is a portable testbed for deploying and implementing city services, including airspace, in real fields using self-standing smart poles.

Fig. 17 Snapshot of self-standing smart point.

Figure 17 demonstrates the self-standing nature of our smart pole in an actual field. The self-standing smart pole in the currently developed version is approximately 8 meters long. It can stand and maintain its upright posture to cancel its tilt by the wind power generated by the eight rotors. Therefore, we can emulate hovering drones at any altitude by placing the experiment items as payloads at any height. In addition, the self-standing smart pole is relatively easy to keep safe compared with real drones, thereby physically supporting them even if an incident occurs.

We conducted wireless communication experiments using self-standing smart poles to develop a sensing system in D2EcoSys. In this experiment, we emulated hovering drones and ground nodes using self-standing smart poles and tripods. We assumed four types of communication scenarios to analyze the difference between the ground and airspace from the viewpoint of radio propagation: (1) ground-to-ground, (2) ground-to-air, (3) air-to-ground, and (4) air-to-air. The experiment was conducted in a multi-purpose court on the Omiya campus of the Shibaura Institute of Technology.

We used GL-AXT1800, which has IEEE~802.11ax (Wi-Fi6, 2.4 GHz band, and 573 Mbps in maximum), as the wireless communication device. In addition, we connected GL-AXT1800 to a Raspberry Pi 4 Model B (8 GB) via a LAN cable as the computational device. GL-AT1800 was mounted at 8 m on a smart pole as a payload to emulate hovering drones, and the Raspberry Pi was placed directly on the ground. To emulate the ground node, we fixed the device 1.5 meters above the ground using a tripod. In this experiment, we generated traffic for throughput measurements using iPerf installed in a Raspberry Pi. We adopted TCP CUBIC, and the TCP maximum segment size (MSS) was set to 1448 Bytes. We generated TCP traffic for the through-

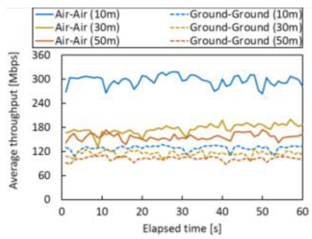


Fig. 18 Time transition of average throughputs of air-to-air and ground-to-ground communications.

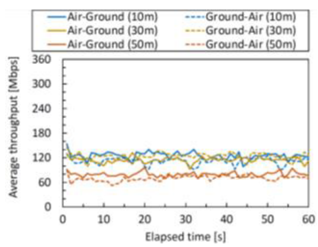


Fig. 19 Time transition of average throughputs of air-to-ground and ground-to-air communications.

put measurements in 60 sec. We changed the measurement distance between the wireless devices from 10 to 50 meters.

Figures 18 and 19 show the time transitions of the average throughputs in ground-to-ground, ground-to-air, air-to-ground, and air-to-air communications. The results for a distance of 10 m between devices indicate that the average throughput of air-to-air communication is significantly higher than that of ground-to-ground communication. In addition, the difference in throughput decreases when the distance increases. The throughputs of ground-to-air and air-to-ground communications were lower than those of air-to-air and ground-to-ground communications. The results include various environmental effects in addition to the settings of experimental devices caused by real fields, such as reflection from the ground. Therefore, we can examine the real performance in various fields using smart poles as hovering drones.

6. Conclusions

In this paper, we introduced an overview of the D2EcoSys project: vision, problem statement, and approach to creating the ecosystem of a co-creation digital twin city with dwellers. The D2EcoSys project adopts decentralized (or Peer-to-Peer-based) participatory sensing to solve the three issues: centralized architecture, incentive mechanism design, and data ownership. These are contained in the previous participatory sensing. Our idea is to keep storing the data in the participant devices and distribute the data via P2P overlay networks. In addition, we assume that the participant devices also offer processing and sensing capabilities. Toward realizing the idea, we demonstrated the recent research results, such as ICN-based service mesh and Name-based efficient content discovery. In addition, we showed the developing testbed environment in KOIL MOBILITY FIELD at Kashiwa-no-ha smart city. The testbed environment can be available for ground and air wireless mesh networks using 60 GHz wireless networks and self-standing smart poles.

In the future, we will establish underlying technologies to meet our ecosystem's requirements, implement the ecosystem's prototype, and demonstrate its performances at Kashiwa-no-ha smart city.

Acknowledgments

This work was partially supported by NICT, grant number 05601, Japan. In addition, we are grateful to BeMap and Advantech Japan for their help with the testbed implementation and the on-site experiments.

References

[1] L. Deren, Y. Wenbo, and S. Zhenfeng, "Smart city based on digital twins," Comput. Urban Sci., vol.1, Article number:4, March 2021.

[2] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," Transdisciplinary Perspectives on Complex Systems, pp.85–113, Springer Cham, 2016.

[3] B.R. Barricelli, E. Casiraghi, and D. Fogli, "A survey on digital twin: Definitions, characteristics, applications, and design implications," IEEE Access, vol.7, pp.167653–167671, Dec. 2019.

[4] T. Deng, K. Zhang, Z.-J. (Max) Shen, "A systematic review of a digital twin city: A new pattern of urban governance toward smart cities" Journal of Management Science and Engineering, vol.6, no.2, pp.125–134, June 2021.

[5] Tomorrow's world today, "Singapore's Digital Twin of Entire Country," Sept. 2022. [online]: https://www.tomorrowsworldtoday.com/2022/09/12/singapores-digital-twin-of-entire-country/

[6] PLATEAU project by MLIT, [online]: https://www.mlit.go.jp/plateau/

[7] NICT Beyond 5G &D Promotion Project, D2EcoSys Project, [online]: https://www.nict.go.jp/collabo/commission/B5Gsokushin/B5G_05601.html

[8] Waseda University, Press release, "Towards realization of co-creating digital twins," [online]: https://www.waseda.jp/top/news/83150

[9] K. Kanai, T. Yamazaki, S. Miyata, H. Kanemitsu, A. Mine, S. Mori, and H. Nakazato, "Research and development of co-creating digital twins using Web3 technologies to accelerate beyond 5G," IEICE Technical Report, CS2022-48, Nov.2022.

[10] Mission of Kashiwa-no-ha Smart City, [online]: https://www.kashiwanoha-smartcity.com/mission/

[11] J.A. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M.B. Srivastava, "Participatory sensing," Center for Embedded Network Sensing, 2006.

[12] T. Campbell, S.B. Eisenman, N.D. Lane, E. Miluzzo, and R.A. Peterson, "People-centric urban sensing," WICON'06, Aug. 2006.

[13] J. Niwa, K. Okada, T. Okuda, and S. Yamaguchi, "MPSDataStore: A sensor data repository system for mobile participatory sensing," ACM SIGCOMM workshop on Mobile Cloud Computing, 2013.

[14] G. Chatzimilioudis, A. Konstantinidis, C. Laoudias, and D. Zeinalipour-Yazti, "Crowdsourcing with smartphones," IEEE Internet Comput., vol.16, no.5, pp.36–44, 2012.

[15] Y. Kaneko, K. Harumoto, S. Fukumura, S. Shimojo, and S. Nishio, "A location-based peer-to-peer network for contextaware services in a ubiquitous environment," IEEE Symposium on Applications and the Internet Workshops, 2005.

[16] V. Jacobson, D.K. Smetters, J.D. Thornton, M.F. Plass, N.H. Briggs, and R.L. Braynard, "Networking named content," ACM CoNEXT,

pp.1–12, Dec. 2009.
[17] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," ACM SIGCOMM Comput. Commun. Rev., vol.44, no.3, pp.66–73, July 2014.
[18] Cloud Native Computing Foundation, [online]: https://www.cncf.io/
[19] K. Kanai, T. Tsuda, H. Nakazato, and J. Katto, "Information-centric service mesh for autonomous in-network computing," ACM ICN 2022, pp.159–161, Sept. 2022.
[20] H. Kanemitsu and H. Nakazato, "KadRTT: Routing with network proximity and uniform ID arrangement in Kademlia," Proc. IFIP Netw. Conf. 2021.
[21] H. Kanemitsu, K. Kanai, and H. Nakazato, "Lookup parameter optimization for Kademlia DHT alternative in IPFS," Proc. IEEE IPDPS2023, May 2023.
[22] P. Maymounkov and D. Maziéres, "Kademlia: A peer-topeer information system based on the XOR metric," Peer-to-Peer Systems, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol.2429, pp.53–65, 2002.
[23] Y. Arakawa and Y. Matsuda, "Gamification mechanism for enhancing a participatory urban sensing: Survey and practical results," Journal of Information Processing, vol.24, no.1, pp.31–38, Jan. 2016.
[24] H. Hotoyama, J. Watanabe, S. Miyata, K. Kanai, and T. Yamazaki, "Modeling and numerical analysis of data transaction by alliance formal game for realization of co-creation digital twin," IEICE Technical Report, IN2022-116, March 2023.
[25] A. Mine, "A blockchain system for IoT data reliability assurance," IEICE Technical Report, May 2023.
[26] S. Mori, "Information-centric wireless sensor networks for smart-city-as-a service: Concept proposal, testbed development, and fundamental evaluation," Proc. IEEE CCNC 2023, pp.945–946, Las Vegas, NV, USA, Jan. 2023.
[27] KOIL MOBILITY FILED at Kashiwa-no-ha smart city: https://www.mitsuifudosan.co.jp/corporate/news/2021/0610_01/
[28] S.G. Kulkarni, M. Arumaithurai, A. Tasiopoulos, Y. Psaras, K.K. Ramakrishnan, X. Fu, and G. Pavlou, "Name enhanced SDN framework for service function chaining of elastic network functions," IEEE Computer Communication Workshops, April 2016.
[29] L. Liu, Y. Peng, M. Bahrami, L. Xie, A. Ito, S. Mnatsakanyan, G. Qu, Z. Ye, and H. Guo, "ICN-FC: An information-centric networking based framework for efficient functional chaining," IEEE ICC, May 2017.
[30] M. Król, S. Mastorakis, D. Oran, and D. Kutscher, "Compute first networking: Distributed computing meets ICN," ACM ICN 2019, pp.67–77, Sept. 2019.
[31] Y. Kumamoto and H. Nakazato, "Implementation of NDN function chaining using caching for IoT environments," ACM CCIoT 2020, pp.20–25, Nov. 2020.
[32] Y. Hayamizu, K. Matsuzono, T. Hirayama, and H. Asaeda, "Design and implementation of ICN-based elastic function offloading network for SFC," IEEE NetSoft, July 2021.
[33] L.A. Wardani, T.M.R. Gias, and D. Kutscher, "Decentralized ICN-based dataflow system implementation," ACM ICN 2021, Sept. 2021.
[34] Cefore: Information Centric Networking Platform, [online]: https://cefore.net/
[35] Kubernetes Blog, The Distributed System ToolKit: Patterns for Composite Containers, [online]: https://kubernetes.io/blog/2015/06/the-distributed-system-toolkit-patterns/
[36] D. Trautwein, A. Raman, G. Tyson, I. Castro, W. Scott, M, Schubotz, B. Gipp, and Y. Psaras, "Design and evaluation of IPFS: A storage layer for the decentralized web," ACM SIGCOMM 202, pp.739–752, Aug. 2022.
[37] testground repository, [online]: https://github.com/testground/testground
[38] libp2p Web site, [online]: https://libp2p.io/
[39] KadRTT as libp2p implementation, [online]: https://github.com/ncl-teu/kadrtt-test-plan
[40] Terragraph, https://terragraph.com/docs/runbook/Overview
[41] mmWave Lab@Kashiwano-no-ha, [online]: https://prtimes.jp/main/html/rd/p/000000075.000018483.html
[42] B.-S. Kim, C. Zhang, S. Mastorakis, M.K. Afzal, and J. Tapolcai, "Guest editorial special issue on information-centric wireless sensor networking (ICWSN) for IoT," IEEE Internet Things J., vol.9, no.2, pp.844–845, Jan. 2022.
[43] T. Yamazaki, S. Miyata, and T. Miyoshi, "A city airspace testbed for drone networks in future smart cities," Advances in Engineering and Information Science Toward Smart City and Beyond, Engineering Cyber-Physical Systems and Critical Infrastructures, vol.5, no.7, pp.141–162, June 2023.

Kenji Kanai received the B.E., M.E. and Ph.D. degrees from Waseda University, Tokyo, Japan, in 2010, 2012 and 2015, respectively. He was an assistant professor at Waseda University until 2017 and a researcher at Waseda Research Institute for Science and Engineering until 2023. He is currently an assistant professor at The University of Tokyo. His research interests include Information Centric Networking, Network Function Virtualization, and Network Orchestration. He is a member of IEICE, IPSJ and IEEE.

Hidehiro Kanemitsu received the B.S. degree in science from Waseda University, Japan, and the M.S. and D.S. degrees in global information and telecommunication studies from Waseda University, Japan. His research interests include parallel and distributed computing, grid, peer-to-peer computing, and web service technology. He is currently a senior assistant professor at the school of computer science, Tokyo University of Technology, Japan. He is a member of the IEEE.

Taku Yamazaki received the B.E. and M.S. degrees in electronic information systems from Shibaura Institute of Technology, Tokyo, Japan, in 2012 and 2014, respectively. He received the D.E. degree in computer science and communications engineering from Waseda University, Tokyo, Japan, in 2017. He is presently an associate professor at Department of Electronic Information Systems, College of Systems Engineering and Science, Shibaura Institute of Technology, Saitama, Japan. His research interests include wireless networks, internet of things, and network security.

Shintaro Mori received his B.S., M.S., and Ph.D. degrees from Kagawa University in 2007, 2009, and 2014, respectively. Since April 2014, he has been with the Department of Electronics Engineering and Computer Science, Faculty of Engineering, Fukuoka University, Japan, where he is currently an assistant professor. His research interests include cross-layer design, wireless sensor networks, and information-centric networks. He is a member of IEEE, IEICE, ISSJ, and RISP.

Aram Mine is the director of web3 division, Gaiax Corporation, Tokyo, Japan. His work focuses specially on blockchain technology and decentralized autonomous organization (DAO) to solve social issues and realize business. He is currently the director of Japan Blockchain Association and a committee member of the Japanese national body of ISO/TC307 Blockchain and distributed ledger technologies.

Sumiko Miyata received the B.E. degree from the Shibaura Institute of Technology in 2007, and the M.E. and D.E. degrees from the Tokyo Institute of Technology in 2009 and 2012 respectively. From 2012 to 2015, she was a research associate at the Kanagawa University. Since 2015, she had been an assistant professor at the Shibaura Institute of Technology. Since 2018, she has been an associate professor at the Shibaura Institute of Technology. Her research interests include mathematical modelling and analysis for QoS performance evaluation, queuing theory, game theory and resource allocation problems in communication networks and information security.

Hironobu Imamura received a B.E. degree from Tokai University in 1983. After graduating from the Faculty of Engineering at Tokai University in 1983, he engaged in computer development at SORD Corporation. After working at a semiconductor trading company and a semiconductor development company, he established HAFT Technology Inc. in 1989. He was involved in the launch of SkyPerfectTV and developed and produced embedded systems for major electronics manufacturers. From 2013, he participated in the launch of several drone manufacturers. In 2015, he founded Drone Works Inc. and undertook the development of various unique drones and mobile robots. In 2021, he successfully developed and flew Japan's first drone powered by a hydrogen fuel cell. Currently, he is involved in the development of waterborne robots using hydrogen fuel cells, last-mile micro-mobility solutions, and provides related consulting services.

Hidenori Nakazato received his B.E. degree in Electronics and Telecommunications from Waseda University in 1982 and his M.S. and Ph.D. degrees in Computer Science from the University of Illinois in 1989 and 1993, respectively. He has experience in industry for more than 15 years and joined Waseda University in 2000. He is a Professor at the Department of Communications and Computer Engineering, Waseda University, Japan. His research interests include information centric networking, cooperation mechanisms of distributed programs, and network QoS control.