IEICE TRANSACTIONS

on Communications

DOI:10.1587/transcom.2023EBP3060

Publicized:2023/10/06

This advance publication article will be replaced by the finalized version after proofreading.

A PUBLICATION OF THE COMMUNICATIONS SOCIETY

The Institute of Electronics, Information and Communication Engineers

Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

Content search method utilizing the metadata matching characteristics of both Spatio-temporal content and user request in the IoT era

Shota AKIYOSHI[†a)], ***Nonmember***, **Yuzo TAENAKA**[††b)], **Kazuya TSUKAMOTO**[†††c)], ***Members***, ***and*** **Myung LEE**[††††d)], ***Nonmember***

SUMMARY Cross-domain data fusion is becoming a key driver in the growth of numerous and diverse applications in the Internet of Things (IoT) era. We have proposed the concept of a new information platform, Geo-Centric Information Platform (GCIP), that enables IoT data fusion based on geolocation, i.e., produces spatio-temporal content (STC), and then provides the STC to users. In this environment, users cannot know in advance "when," "where," or "what type" of STC is being generated because the type and timing of STC generation vary dynamically with the diversity of IoT data generated in each geographical area. This makes it difficult to directly search for a specific STC requested by the user using the content identifier (domain name of URI or content name). To solve this problem, a new content discovery method that does not directly specify content identifiers is needed while taking into account (1) spatial and (2) temporal constraints. In our previous study, we proposed a content discovery method that considers only spatial constraints and did not consider temporal constraints. This paper proposes a new content discovery method that matches user requests with content metadata (topic) characteristics while taking into account spatial and temporal constraints. Simulation results show that the proposed method successfully discovers appropriate STC in response to a user request.

key words: *Internet of Things, Cross-domain data fusion, Content search*

1. Introduction

With the development of IoT technology [1], more than 40 billion IoT devices are expected to be connected to the network by 2025, dynamically generating an even greater variety of content. It is also expected that 30 percent of this content will become real-time data [2]. In this case, depending on the type (fixed or mobile) and operation of the IoT device, the data would have various characteristics, such as data collection interval, data collection areas, data volume, and data availability. These IoT data must be processed in real-time in the edge cloud.

We proposed a new information platform, called Geo-Centric Information Platform (GCIP) [3], which collects, processes, and distributes IoT data (i.e., realizes IoT data fusion) in a geolocation-aware manner. As shown in Fig. 1, GCIP consists of nested meshes mapped to geographic locations (longitude and latitude), and each cell in the mesh is assigned a unique ID. By including this ID in the IP address and routing it, IoT data generated within each cell can be stored for each cell. Since each cell also consists of an edge router and multiple servers to perform IoT data fusion in its geographic neighborhood, content specific to that cell can be generated.

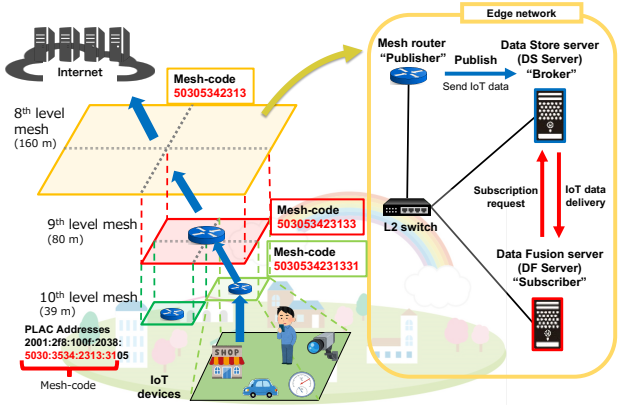


Fig. 1 Assumed environment for GCIP

In the GCIP, two types of servers, a data store server (DS server) and a data fusion server (DF server), are deployed in a cell. The DS server collects all IoT data generated in the corresponding cell, and the DF server uses this IoT data and generates spatio-temporal content (STC) in the cell, which is local to the mesh region.

In such cases, users cannot know in advance "when," "where," or "what" content is being generated because the location, type, and timing of content generation change dynamically according to these data characteristics. This makes it difficult for users to directly search for specific content they have requested using content identifiers (domain name of URI or content name). So, there is a need for new content discovery methods that take into account spatial and temporal constraints that do not use content identifiers.

In our previous work [4], we proposed a spatially constrained STC discovery method that does not use content identifiers by matching user requests with IoT data usage statistics for each DF server. Although this method can select the DF server having the largest amount of STC based on geographic space, a DF server that has less STC is never

†The author is with Kyushu Institute of Technology
††The author is with Nara Institute of Science and Technology
†††The author is with Kyushu Institute of Technology
††††The author is with City College of New York
a) E-mail: akiyoshi.shota190@mail.kyutech.jp
b) E-mail: yuzo@is.naist.jp
c) E-mail: tsukamoto@csn.kyutech.ac.jp
d) E-mail: mlee@ccny.cuny.edu

selected even if that server has some appropriate STC for the user request. In addition, STC has an availability period (time constraint), but this was not taken into account.

In this paper, we propose an extended STC discovery method that takes into account spatial and temporal constraints based on the previous study. The proposed method uses cosine similarity to discover DF servers that have a large amount of STC that is appropriate for the user request and that will also remain available. This method makes it possible to provide content that is spatiotemporally fresh and useful to the user.

The remainder of the present paper is organized as follows. Section 2 introduces related research on content retrieval, and Section 3 gives an overview of our previous research. Section 4 describes the proposed method, and Section 5 describes another method for comparison, evaluation metrics, and simulation results. Finally, Section 6 presents a summary of our paper.

2. Related Work

In this section, we review existing content retrieval methods. In the TCP/IP protocol suite used on the current Internet, users can be made aware of content by a domain name. An IP address is required to connect to a specific host with the content of this domain name. In other words, it is assumed that the domain name of the content is known before the connection is made. This type of network architecture is called Host-Centric Networking (HCN) [5].

HCN is easy to connect to when the recipient can be identified in advance, such as in the case of e-mail. However, it was not easy to find specific content from a huge amount of content, that is, content searching tended to be difficult. This problem has now been solved with the development of the Domain Name System (DNS) [6–9] and search engines [10]. The DNS takes the part that specifies the host from the URL specified by the user and converts the hostname into an IP address. By sending a content request to this IP address, the user can easily retrieve the content. However, it is still difficult for people to know and keep track of all the content (domain names), so to combat this issue, search engines (Yahoo!, Google, etc.) have been developed. This makes it easier to retrieve content as users can simply search for the content that they want using a list search or keyword search.

However, in a content-centric world where large amounts of data are being generated and consumed, as has been the case in recent years, a content placement system such as HCN is hardly suitable. This is because users are not interested in where the content is retrieved from (content location), but rather in being able to retrieve the content itself more quickly. As of 2022, 500 hours of video are uploaded to the platform every minute on YouTube, and people watch one billion hours of video every day [11]. In this scenario, having multiple communication servers is more likely to contribute to load balancing and lower latency than having a single server, so having a fixed (HCN) server is not optimal.

Therefore, around 2010, a new type of network technology known as Information-Centric Networking (ICN) [12–14] was proposed. Named Data Networking (NDN) [15] and Content-Centric Networking (CCN) [16] are two of the well-known network architectures in this category. In ICN, users can retrieve desired content from the network by directly specifying the name of the content to be retrieved instead of the IP address of the destination host. This prevents the name resolution overhead at the start of communication when using DNS. This is also expected to improve the response time and reduce the communication load on a server. However, ICN content is given unique names just like IP addresses in HCN, and thus cannot be retrieved without knowing the name of the content. This search method is effective when the content is available long after it has been deployed, in any location, and is provided on an ongoing basis.

However, in a situation (GCIP environment) where new content is being generated every moment by a large amount of IoT data in the IoT era, content retrieval using content identifiers becomes extremely difficult because users cannot know in advance ”when,” ”where,” or ”what type” of content is being generated. Reference [17] summarizes existing studies that focus on content retrieval (location-based [18], metadata-based [19], and event-based [20]). These methods make independent use of the time, location, and content factors that must be considered in a GCIP environment. Also, all previous content search methods use content identifiers to search.

In this study, we propose a new content retrieval method that does not use content identifiers with the IoT era in mind. This is achieved through a topic-based search method that specifies searches by breaking down content into elements called topics. The proposed method enables content searching even when the user does not know the content name. Additionally, even when the content name is not related to the topic, if the content was created using IoT data on the specified topic, it can be retrieved as related to the topic.

3. Geolocation Centric Information Platform [3]

This section describes the conceptual design of GCIP (Fig. 2) and its key technologies. Procedures of GCIP consist of the following 4 steps [21]. First, IoT data sent from IoT devices within a certain geographic area is replicated by the intermediate router(s) and aggregated by geography (Step 1). Next, the replicated data is forwarded to a proximity edge server with analysis/processing capabilities (Step 2). The server(s) processes the collected data and generates some STC (Step 3). Finally, the server(s) transmits the generated STC to the user(s) (Step 4).

First, we here describe the key technologies that make STEP 1 possible [22]. To collect data based on geographic location, a unique ID embedded in the network address is used to configure a transmission path with a hierarchical mesh structure network topology as shown in Fig. 1. The geographic space is divided into a hierarchical mesh based

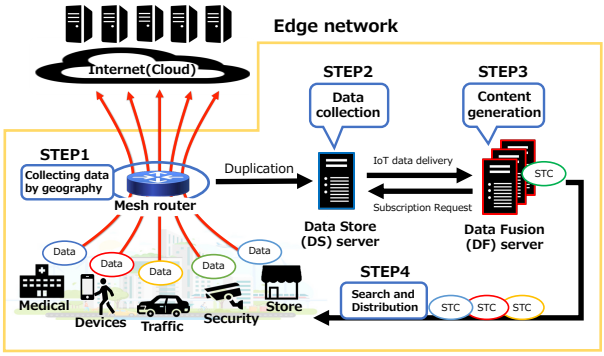


Fig. 2 Conceptual design of GCIP

on latitude and longitude, and a unique ID, called a mesh ID, is assigned to each cell (the size of the smallest mesh area is a square area with 39 [m] on a side according to the Open-i area [23] expansion rule). By including the mesh ID within the IPv6 network address in which the communication infrastructure is prevalent, the network covering the cell can be accessed. The length of this mesh ID increases as the area decreases. Each intermediate router is then assigned a unique mesh code so that the router can identify the geographic area to which it belongs and handle all IoT data containing the same mesh code.

Second, we describe the key technologies that make STEP 2 possible. In each mesh on the GCIP, DS servers are supposed to be installed by local governments, such as prefectures and municipalities, and DF servers are installed by content providers who want to provide STC to users in that region. In this study, we assume that there are a DS server and several DF servers in a mesh of the GCIP. A single DS server manages all IoT data collected in a physical space delimited by a certain latitude and longitude. In other words, all data in a particular DS server is associated with a particular physical area (mesh).

Next, the key technologies that make STEP 3 possible are described here. In order to make STC generation occur asynchronously, we use Publish/Subscribe (Pub/Sub) communication (Fig. 1). Here, the mesh router is called the Publisher, and the DS server is the Broker. The DF server is the Subscriber. The mesh router duplicates all data sent from IoT devices to a particular cloud server (original destination) along the way and publishes these data to the single DS server with a topic indicating the type of data. At STC generation, a DF server sends a subscription request to the DS server specifying multiple topics and processes the received data to generate an STC item. At this time, the availability period for each STC item is set by the DF server. In the present study, we assume that one STC item is generated by a set of data collected in one subscription request. Note that even if a subscription request consists of the same combination of topics, a different STC item may be generated, but the DS server is not concerned about this point.

From the next section, we explain the method of matching the STC generated from the IoT data collected on a geographic basis in STEPs 1 to 3 with the user requests in STEP 4.

4. Matching-based STC Discovery

4.1 Conceptual design of the matching search method

Since each DF server is managed by a different operator and the type and timing of content generation on each DF server varies, users cannot know what STC is being generated or when and where this generation occurs, and thus cannot directly request any DF server for any STC. In this circumstance, since it is also difficult for users to specify the name of the content or explicit keywords, a new search method is required. Therefore, we focus on the fact that the DS server receives subscription requests from all DF servers in a cell at the time of STC generation (STEP 3 in Fig. 2) and thus can use its statistical information.

The following is an overview of the matching-based search. The user first sends a search request to the DS server, which acts as an anchor point in each mesh because the DS server has all the data in each mesh and has the subscription statistics. When the DS server receives the search request, the server tries to match the subscription statistics for all DF servers with the user request and chooses an appropriate DF server for the request. The DS server then forwards the search request to the DF server, and the DF server sends the user STC that is appropriate for the user request. In the following, we describe the previous method and its problems, and then explain the user requirements and matching procedures for the proposed method.

4.2 Previous search method [4] and remaining problems

In the previous search method, a user is supposed to specify several topics, which are highly related to the interest of the user. Each topic has a priority of 0 to 100 but the sum of all topic priorities is 100. This request is sent to the DS server, and the DS server chooses the DF server that is expected to have the largest amount of STC matching the user request. In order to calculate the expected value of the amount of STC, E_i at DF server i, if we assume that the combination of topics that satisfy the user request is c ($c \in C$), then the request probability of topic j could be $P_{i,j}$, and the probability that DF server i has combination c is $G_i(c) = \prod P_{i,j}$. Using these definitions, the expected value of the amount of STC on DF server i can be expressed as $E_i = \sum_{c \in C}(G_i(c) \times N_i)$. N_i is the subscribe times of the DF server.

Although this method can select a DF server having the largest amount of STC, it has two problems. One is that a DF server that has less than the largest amount of STC can never be selected, even if that server has appropriate STC for the user request. This is due to the fact that only one DF server is selected, and that server is expected to have the most STC that matches the user request. The other issue is that since all STC items have an availability period, this should be taken into account in the search procedure, but it is not. Without

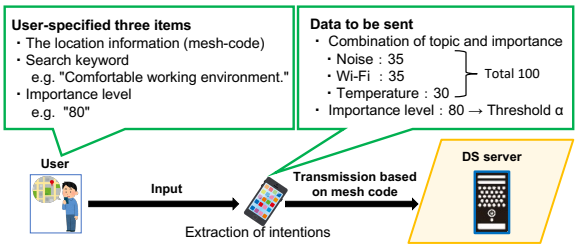


Fig. 3 Overview of sending search requests

DF server	List 1 Amount of appropriate STC	List 2 Available period	Score List 1+List 2
server S_1	3rd.	3rd.	6
server S_2	4th.	4th.	8
server S_3	2nd.	1st.	3
・	・	・	・
・	・	・	・
Optimal DF server			
server S_I	1st.	2nd.	3

Fig. 4 Example of determining the optimal DF server

this consideration, a selected DF server might have only old (not useful) STC.

4.3 Proposed method

In the proposed method, as shown in Fig. 3, a user specifies three types of information for STC discovery: location information for the target area, the search keywords for the desired content, and the importance level, which indicates how much of the user's intent is included in the search. The importance level, a number from 0 (allowing anything) to 100 (strictly the same), indicates the strength for the user's intention on particular topics in percent. For example, specifying 80 will select STC that contain at least 80% of the keywords (topics) requested by the user, while others (20% or less) may yield unintended new information. The search keywords are translated (or decomposed) to topics by using any intention extraction technique [24] [25], which is beyond the scope of this paper. At this time, a priority, a number from 0 to 100, is also assigned to each topic. The sum of priorities for all topics is 100. Once the information is received as a request, the DS server tries to identify an appropriate DF server by the proposed two-stage search method and then forwards the request based on the information. The definition of the optimal DF server is described in Section 4.3.1. The first stage of selecting several candidates for an optimal DF server is described in Section 4.3.2, and the second stage of selecting the optimal DF server is described in Section 4.3.3.

4.3.1 Defining the optimal data fusion (DF) server

Since the proposed method takes the freshness and amount of STC into account in STC search, we need the definition of the optimal DF server based on these two factors. We define the optimal DF server as a server that has the largest amount of STC, that matches the topics given in a user request, and that has a longer remaining availability period. A server having a large amount of fresh STC is more beneficial to users than a server having a large amount of old STC, which is sometimes identified by the method described in Section 4.2.

The formal definition is as follows. We make two lists in which DF servers are sorted in the order of the amount of appropriate STC and the total availability period for all appropriate STC, respectively. The score for each DF server is calculated by the sum of two numerical values, indicating their order on these lists. The DF server having the smallest score is treated as optimal. Although this definition of the optimal DF server is useful to determine a theoretically optimal server, in practice, nobody has a global view of all DF servers. This is why the proposed method tries to identify an optimal server by using statistics on the subscription requests from every DF server, as will be described in the next section. If the sum of the ranks is the same, then the DF server with the largest amount of STC satisfying the user requirements is defined as the optimal DF server (Fig. 4). In this way, servers with a large amount of only old STC, or servers with a small amount of STC but a very long availability period for one piece of STC, are not selected, and instead a server with a large amount of STC and STC with a long availability period for the entire STC will be determined to be the optimal server.

4.3.2 Stage 1: Matching algorithm for selecting several candidates of optimal DF server

Figure 5 shows the matching procedure for the proposed method. In order to identify an optimal DF server, a DS server estimates an optimal DF server by matching the subscription statistics on DF servers to the user request. Specifically, a DS server counts the number of subscription requests for each topic sent from each DF server and calculates the ratio of subscriptions for each topic in all subscriptions. A higher ratio for a topic indicates a DF server is more likely to have a larger amount of STC on that topic. In contrast, since a user request includes several topics each having a priority value, this can be treated such that the user expects STC composed in part from topics with the ratio of the priority value. From this similar context, a DS server that has information about the subscription request and the user request matches these requests to find an optimal DF server.

In order to perform matching, we use cosine similarity to evaluate the similarity of the topic composition in the subscription of a DF server and user request. The DS server keeps the combination of topics subscribed to by each DF server, the last subscription time for each topic from each DF servers, the subscription interval for each subscription with the same topic combination, and the total number of subscriptions. In order to describe the procedure identifying an optimal DF server, we use the following notation for

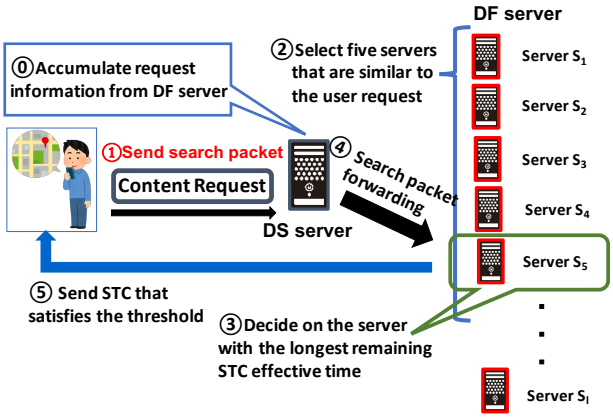


Fig. 5 Matching procedure for the proposed method

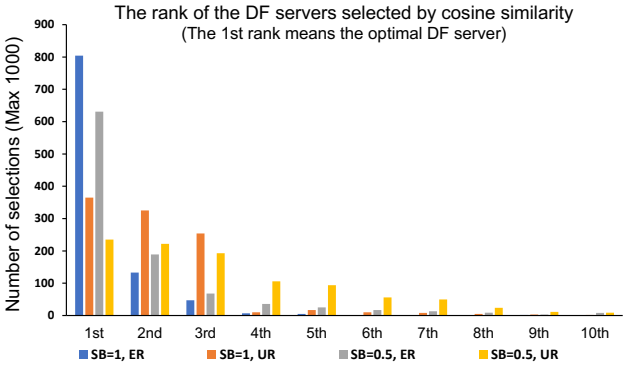


Fig. 6 The rank of the DF servers selected by cosine similarity

subscription information for the DF server and user request:

- Total number of DF servers in the mesh: L
- All topic combinations subscribed to by DF server i: $C_i = \{c_{i1}, c_{i2}, .., c_{im}, ..., c_{iM}\}$
- c_{im} containing topic j in C_i: C'_j
- Last subscription time for c_{im}: $t_{c_{im}}$
- Subscription interval for c_{im}: $i_{c_{im}}$
- Total number of subscriptions for c_{im}: $n_{c_{im}}$
- Total number of subscriptions for STC that satisfy the user request: n_{sum}

Next, we calculate the ratio of the combination c_{im} to the subscriptions of one DF server. We set the importance level specified in a user request as a threshold α and select only topic combinations c_{m_α} where the total value of the priority in the user request exceeds α. That is, the topic combinations depend on the importance level, α, which could be that of only one topic, even if a user request includes several topics. This is to obtain a search result involving related information. We define the weight for each element in c_{im_α} as $w_{c_{im}} = n_{c_{im}}/n_{\text{sum}}$. Then, we define the weight w_j of topic j as the sum of the $w_{c_{im}}$ of the elements in C'_j. The normalized vector of weights for each topic is defined as the weight vector W of the DF server. Finally, we define W_U as the weight vector of the DF server when the user specifies the importance of a topic. Using these vectors, we calculate the cosine similarity as in Eq. 1.

$$CS_l = CosSim(W, W_U) \tag{1}$$

Next, we will explain how to use cosine similarity. The larger the value of the cosine similarity, the more optimal the DF server is considered to be. Therefore, the DF server with the largest CS_l calculated using Eq. 1 is estimated to be the optimal DF server. Fig. 6 shows the ranking of servers selected only by cosine similarity in the preliminary experiment. From the simulation results, it is clear that the server with the highest cosine similarity is not necessarily the optimal DF server. This is because the subscription condition does not include a time factor, and the optimal DF server selected based only on cosine similarity does not necessarily have a long effective STC time. In other words, it is difficult to estimate the optimal cross-disciplinary collaboration server based solely on the size of the cosine similarity. Therefore, in Stage 1, the DF servers in the top 90% selected based only on cosine similarity are selected as candidates for the optimal DF server in each environment with different numbers of servers and content subscription bias. In this way, it is possible to narrow down the heuristically optimal server candidates to half of the total in Stage 1. Since the number of servers selected at this time depends on the environment, it is necessary to investigate the optimal parameters for each environment.

4.3.3 Stage 2: Optimal DF server selection

The next step is to select one of the candidates chosen in Section 4.3.2. We use the availability period to select a candidate. The DF server with the largest median STC availability period is defined as the server with the longest availability period. This definition eliminates the possibility of selecting a server having old STC. However, since the DS server cannot know the availability period for each STC, it uses the value of the Poisson distribution $p_{c_{im}}$ of the mean availability period λ to estimate the distribution. Specifically, the remaining availability period for the combination c_{im} is calculated as $e_{c_{im}}$ in Eq. 2. The arrival time of the search request of the user is set to t_{now}.

$$e_{c_{im}} = p_{c_{im}} - (t_{now} - t_{c_{im}}) \tag{2}$$

Let U_{S_l} be the median of the $e_{c_{im}}$ aggregate of DF servers S_l. Among all the DF servers in the mesh, the server with the largest U_{S_l} has the highest probability of being the optimal DF server, so the DS server forwards the search request of the user and performs the search.

The DF server that receives the forwarded request searches for STC composed of topic combinations c_{m_α} and then returns all found STC to the user.

Table 1 Simulation cases

	biased	unbiased
User request	Explicit request (ER)	Unclear Request (UR)
Subscription request	SB=1	SB=0.5

5. Performance Evaluation

5.1 Simulation environment

Our simulation environment is shown in Fig. 7. There are 10 DF servers, and each DF server is supposed to request many subscriptions to a particular topic. In the cloud environment, it is expected that servers in the order of 100 or 1000 will be deployed, but in the edge cloud environment, the number of servers in an edge network will be limited to 1/10 to 1/100 of those. Therefore, we decided to deploy 10 DF servers for this verification.

In this study, to focus on matching the bias between user requests and subscription requests at the time of STC generation. At this time, the topics included in the user's search and the topics subscribed to by the DF server can be roughly classified into two types as shown in Table 1: one in which certain topics are considered as an important (biased) and the other in which certain topics are not considered as an important (unbiased). The bias of user requests is defined as explicit request (ER) and unclear request (UR), and the subscription bias at the time of STC generation on the DF server is defined as SB. To verify the impact of the presence or absence of these biases on the proposed method, we investigate the performance of the proposed scheme under cases (described below) where the two sorts of bias are happened.

First of all, to confirm for differences due to bias in the topics requested by the user, we do this with ER, where there is a large difference in the importance level of each topic, and UR, where there is little difference in the importance level of each topic. If three topics are requested, an example of ER would be (x,y,z)=(80,15,5), and an example of UR would be (x,y,z)=(35,35,30). In the simulation, these parameters were used to obtain the characteristics of the method under biased conditions. However, the range of these values is not limited and any combination is available such that the total is 100. Since it is difficult to validate all patterns, several combinations of characteristics were selected and set as examples for ER and UR.

Second, we use subscription bias (SB), which represents the bias of topics requested from the DF server to the DS server. As a DF server creates a variety of content in parallel, the topics required for creation may differ depending on what is being created. In GCIP, the DS server does not manage the contents of the DF server itself, because we assume an environment in which the contents themselves are highly constrained in space and time. Therefore, the DS server judges what STC it is generating based solely on the statistics of requests from the DF server. In this study, we investigated the effectiveness of the proposed method by assuming

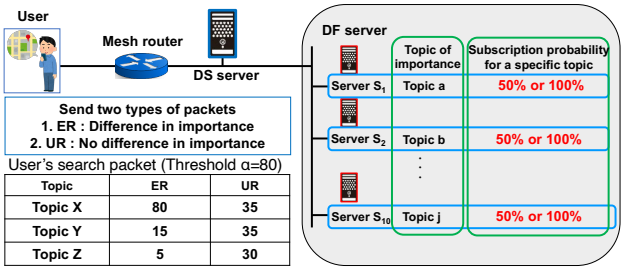


Fig. 7 Simulation topology

Table 2 Simulation parameters

Parameter name	Parameter value
Number of DF servers	10
Number of content	100 [piece/unit]
Topic Type	10 [types]
Number of topics linked	2∼5 (randomly determined)
STC mean period to available	10,30,60 [minutes]
Threshold α	80
Mean period to available λ	10,30,60 [minutes]

two patterns of SB as the statistical information available to the DS server: one in which a particular topic is always used 100% of the time (SB=1) and the other in which a topic is used 50% of the time (SB=0.5).

Thus, a total of four cases are simulated. In the simulation, STC is generated for 10 minutes, and then a search request is sent for evaluation. We set the parameters as shown in Table 2. The user sends ER and UR to the DS server 1,000 times each.

5.2 Evaluation index

Four indexes are used to evaluate the results. The first is the estimation accuracy, which indicates whether the DF server estimated by each method is the optimal DF server. The second is the number of appropriate STC obtained from the DF servers estimated by each method. The third is the distribution of the remaining availability period for the obtained STC (Eq. 2). The fourth is the unconscious contents ratio (UCR), which indicates how many topics that the user has not specified are included. Note that, we call the condition in which the obtained STC is composed of exactly the same topics as included in a user request "em". Furthermore, we may say that em+1 when the amount of STC contains one topic other than the user request, and em+2 when the amount of STC contains two extra topics. The UCR is calculated using Eq. 3. C_{total} is the total number of STC retrieved by the user and $C_{complete\_match}$ is the number of STC created with exact same topics with the user request.

$$UCR = \frac{C_{total} - C_{complete\_match}}{C_{total}} \times 100 \tag{3}$$

We use two comparison methods to evaluate the effectiveness of the proposed method. For comparison method 1, we use the method of previous studies described in Section

4.2, and this method is referred to as expected value-based (EV-based). This method uses the number of STC matching the user's request and the number of subscriptions to each topic to select the DF server that is expected to have the largest amount of STC matching the user request. The expected value E_i at DF server i is shown in Eq. 5. Let c ($c \in C$) be the combination of topics satisfying user requests, $P_{i,j}$ be the request probability of topic j at DF server i, and $G_i(c)$ (Eq. 4) be the probability that DF server i has combination c.

$$G_i(c) = \prod P_{i,j} \tag{4}$$

$$E_i = \sum_{c \in C} (G_i(c) \times N_i) \tag{5}$$

Comparison method 2 uses only the cosine similarity to select the appropriate DF server, and this method is referred to as cosine similarity-based (CS-based). The difference with the proposed method is that it does not take into account the remaining availability period. In this method, the server with the largest value of CS_l derived by the proposed method is selected as the optimal server.

5.3 Results and discussion

Figures 8 and 9 show the estimation accuracy of the DF server in the experiments in four cases in which user request (ER/UR) and subscription bias (SB) 1.0/0.5 are combined. These figures include the result of the proposed method, the EV-based method (comparison method 1), and the CS-based method (comparison method 2), respectively. The order of the identified DF server, denoted as 1st, 2nd, and 3rd, is in the order of the score, which is used for the definition of the optimal DF server in Section 4.3.1. Here, 1st indicates that a method found the optimal DF server.

These figures show that the proposed method is the most accurate in estimating the optimal DF servers in all four cases. For the case in which both the user request and SB are highly biased (ER, SB=1), the proposed method is able to estimate the optimal DF server (1st.) with 67% accuracy and the top three DF servers with 92% accuracy. In the same case (ER, SB=1), the accuracy of estimating the top 3 DF servers is 5.2% for the EV-based method and 77.2% for the CS-based method, indicating that the proposed method can improve by 86.8% and 14.8%. On the other hand, even when both user requests and SB have small bias (UR, SB=0.5), the proposed method is able to estimate the optimal DF server (1st.) with 47% accuracy and the top three DF servers with 81% accuracy. In the same case (UR and SB=0.5), the next highest estimation accuracy of the optimal DF server (1st.) is for the CS-based method, with a result of 20%. In the same case (UR, SB=0.5), the estimation accuracy of the top three DF servers was 14.9% for the EV-based method and 49.9% for the CS-based method, indicating that the proposed method can improve by 66.1% and 31.1%.

These results show that the estimation accuracy was higher for cases with a high user request bias and subscription bias (ER and SB=1), and lower for cases with a low user request bias (UR and SB=0.5). This is because the smaller the subscription bias, the less biased the topics that the DF server considers important, and as a result, many DF servers have topics equally close to user requirements. This makes it difficult to distinguish between the best DF server (first place) and other DF servers (second and higher) (no difference in cosine similarity values), resulting in lower estimation accuracy.

To summarize these results, the proposed method employs a two-stage algorithm that selects the server with the most STC (Stage 1) and the server with the longest remaining availability period (Stage 2), thus allowing the optimal server to be selected. On the other hand, the EV-based system does not consider the remaining available period and searches for STC that perfectly match the user request, resulting in low estimation accuracy. The CS-based system also does not consider the remaining available period and results in low estimation accuracy.

Next, Tables 3 and 4 summarize the average value of the amount of STC acquired by users in four cases. For the case in which both the user request and SB are highly biased (ER, SB=1), the proposed method (56 total) and CS (57 total) obtain more STC than EV (12 total). This is true in all four cases. This is because EV does not consider em+1 and em+2, and thus cannot estimate the DF server that maximizes the STC obtained by including em+1 and em+2.

Figures 10 and 11 show the results of the remaining availability period of the STC acquired by users. These figures include the result of the proposed method and the CS-based method. The difference in median availability period between the proposed method and the STC obtained by CS was approximately 15 minutes for SB = 0.5 and UR with small bias and approximately 17 minutes for SB = 1 and ER. The proposed method can provide STC in all cases with a longer availability period than the methods with CS and EV. To find the reason for these results, we considered the availability period for the STC. Table 5 and 6 summarize the distribution of availability periods for the number of STC obtained for ER and UR. From these results, it is evident that the CS method obtains STC so that each availability period is equal. On the other hand, the proposed method acquires STC so that the availability period of 60 minutes is larger and the availability period of 10 minutes is smaller. Therefore, the median availability period for the acquired STC is larger in the proposed method (because the proportion of STC with longer availability periods is larger among the acquired STC). It can therefore be said that the proposed method can provide users with more STC with longer availability periods.

Table 7 shows the UCR results. In all methods, we can obtain STC that includes topics not yet specified by the user, which can give the user new insights. This is beneficial because it allows users to obtain additional information on the

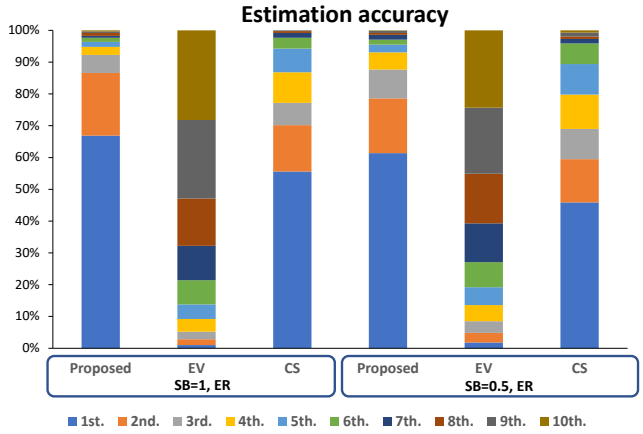


Fig. 8 Estimation accuracy in explicit request (ER)

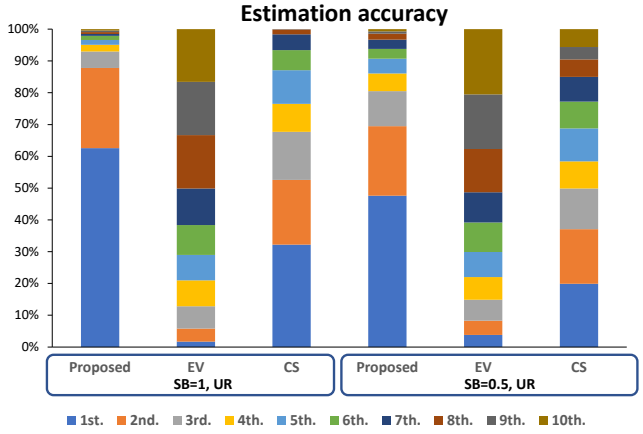


Fig. 9 Estimation accuracy in unclear request (UR)

topic they are searching for. In addition, in the EV, since the server with the highest ranking is not selected, it is not possible to obtain STC that perfectly matches the user requirements, but only STC that contains topics that are not specified by the user. This result depends on the importance α. For α=100, the UCR is expected to be 0, since only STC that perfectly matches the user's intentions are selected. On the other hand, as α approaches 0, the UCR increases because all STC are selected regardless of the user's intentions. In our simulation, α=80, we obtain a UCR of 20% while taking into account 80% of the user's intentions. we need to investigate the effect of changing α, i.e., the effect on the number of UCR, but it changes depending on the content creation situation. That is, we need to design realistic data-fusion environment where a lot of real content are made autonomously and dynamically time by time. That is why, as the current paper was aiming a first trial to search for uncertain contents in IoT data fusion situation, our focus is the matching mechanism and the evaluation of the mechanism itself.

Table 3 Number of STC acquisitions in explicit request (ER)

	ER(SB=1)			ER(SB=0.5)		
	em	em+1	em+2	em	em+1	em+2
Proposed method	4	14	28	3	8	15
EV-based	0	2	9	1	4	13
CS-based	5	18	34	3	9	15

Table 4 Number of STC acquisitions in unclear request (UR)

	UR(SB=1)			UR(SB=0.5)		
	em	em+1	em+2	em	em+1	em+2
Proposed method	4	15	29	3	8	14
EV-based	0	3	11	1	4	13
CS-based	8	20	32	5	10	12

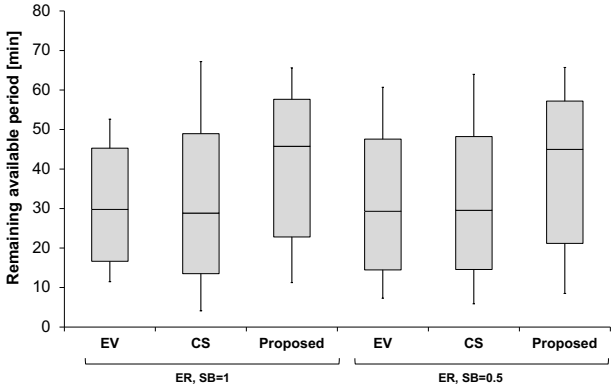


Fig. 10 Remaining availability period of acquired STC in ER

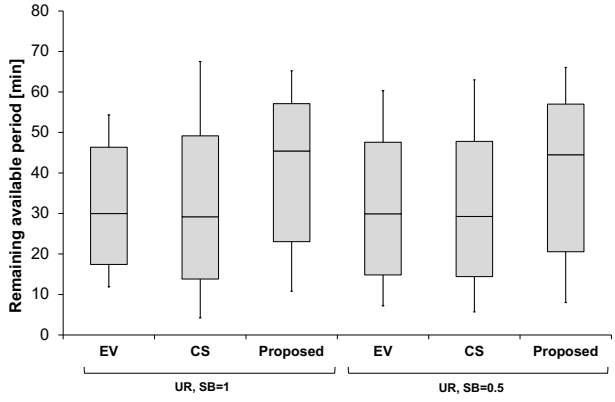


Fig. 11 Remaining availability period of acquired STC in UR

Table 5 Distribution of availability period for the number of STC obtained in explicit request (ER)

	ER(SB=1)			ER(SB=0.5)		
Availability period [min]	10	30	60	10	30	60
CS-based	19	19	19	9	9	9
Proposed method	8	10	16	5	6	11

6. Conclusion

In the GCIP, users cannot know when and where any STC

Table 6 Distribution of availability period for the number of STC obtained in unclear request (UR)

	UR(SB=1)			UR(SB=0.5)		
Availability period [min]	10	30	60	10	30	60
CS-based	19	20	20	9	9	9
Proposed method	8	10	16	5	6	12

Table 7 Unconscious contents ratio (UCR)

	ER		UR	
	SB=1	SB=0.5	SB=1	SB=0.5
Proposed method	94.8	95.6	90.4	85.8
EV-based	100	95.5	100	94.0
CS-based	94.8	97.3	86.8	85.8

is generated, nor can they directly request any DF server to search for STC. Therefore, we proposed a matching approach for STC searches that satisfies the user request by focusing on the similarity between the subscription statistics on DF servers and the user request and the availability period of STC. The simulation results showed that the user can obtain fresh STC using the proposed method. In the future, we plan to further consider different realistic content creation situations and methods to improve retrieval accuracy by estimating the amount of STCs generated from the transmission interval of the same subscription.

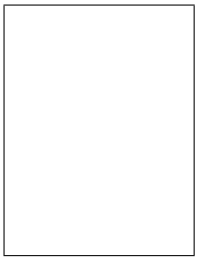
Acknowledgments

The present study was supported by JSPS KAKENHI Grant number JP21H03430, and the commissioned research No.05501 by NICT, Japan.

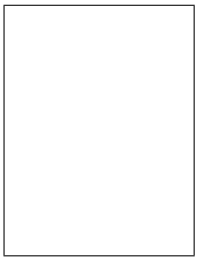
References

[1] A. Al-Fuqaha, et al. "Internet of Things: A Survey on Enabling Technologies, Protocols and Applications," IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347–2376, Jun. 2015.
[2] S. Condon, "By 2025, nearly 30 percent of data generated will be real-time, IDC says," 2022, https://www.zdnet.com/article/by-2025-nearly-30-percent-of-data-generated-will-be-real-time-idc-says/
[3] K. Tsukamoto, et al. "Geolocation-centric Information Platform for Resilient Spatio-temporal Content Management," IEICE Trans. Commun., Online ISSN 1745-1345, Print ISSN 0916-8516, Sep. 2020.
[4] K. Nagashima, et al. "Matching based content discovery method on Geo-Centric Information Platform," INCoS 2020, vol. 1263, pp. 470–479, Sep. 2020.
[5] F. Karim, A. Aman, R. Hassan, K. Nisar, "A Survey on Information-Centric Networking with Cloud Internet of Things and Artificial Intelligence", Wireless Communications and Mobile Computing, vol. 2022, Article ID 7818712, 11 pages, 2022.
[6] RFC 2181, Clarifications to the DNS Specification, Internet Standard, 1997.
[7] Public DNS Server List (http://public-dns.tk/,).
[8] P. V. Mockapetris, RFC 1034, Domain Names-Concepts and Facilities 1987.
[9] P. V. Mockapetris, RFC 1035, Domain Names-Implementation and Specification 1987.
[10] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," Computer Networks and ISDN Systems Vol. 30, No. 1–7, pp. 107–117, Apr. 1998.
[11] J. Wise, "How much data is created every day in 2022," 2022, https://earthweb.com/how-much-data-is-created-every-day/
[12] G. Xylomennos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, G. Polyzos, "A Survey of Information-Centric Networking Research," IEEE Communication Surveys and Tutorials, no. 99, 26 pages, Jul. 2013. DOI: 10.1109/SURV.2013.070813.00063!
[13] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, B. Ohlman, "A Survey of Information-Centric Networking," IEEE Communications Magazine, vol. 50, no. 7, pp. 26–36, Jul. 2012. DOI: 0.1109/MCOM.2012.6231276!
[14] M. Bari, S. Chowdhury, R. Ahmed, R. Boutaba, B. Mathieu, "A Survey of Naming and Routing in Information-Centric Networks," IEEE Communications Magazine, vol. 50, no. 2, pp. 44–53, Dec. 2012. DOI: 10.1109/MCOM.2012.6384450!
[15] L. Zhang, et al. Named data networking. ACM SIG- COMM CCR, vol. 44, no. 3, pp. 66–73, 2014.
[16] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content," Proc. ACM CoNEXT 2009, pp.1–12, Dec. 2009.
[17] S. Pattar, et al. "Searching for the IoT resources: fundamentals, requirements, comprehensive review, and future directions," IEEE Commun. Surv. Tutorials vol. 20, pp. 2101-2132 (2018).
[18] S. Mayer, D. Guinard, V. Trifa, Searching in a web-based infrastructure for smart things. In: 2012 3rd IEEE International Conference on the Internet of Things, pp. 119–126, Oct. 2012.
[19] S. Mayer and D. Guinard, An extensible discovery service for smart things. In: WoT 2011: Second International Workshop on the Web of Things, pp. 1–6 Jun. 2011.
[20] A. Pintus, D. Carboni, A. Piras, Paraimpu: a platform for a social Web of Things. In: Proceedings 21st International Conference on Companion World Wide Web (WWW Companion), pp. 401–404, April 2012.
[21] K. Nagashima, et al. "Experimental Evaluation of Publish/Subscribe-based Spatio-Temporal Contents Management on Geo-Centric Information Platform," NBiS-2019, vol 1036
[22] H. Tamura, Program for determining the ip address based on the position information, apparatus and method, The publication of Japanese Patent No. JP6074829B1, 2017-2-8.
[23] NTT DoCoMo : Open iArea guideline, https://www.nttdocomo.co.jp/binary/pdf/service/developer/make/content/iarea/ domestic/open-iarea.pdf (in Japanese)
[24] M. Kurihara, et al. "Extracting Local Resident Demands per Region Using Twitter,". JSAI2015, vol. 29, pp1-4, 2015
[25] K. Kaneko, et al. "Purchase Intent Extraction and Intent Hierarchy Construction from Customer Reviews,". JSAI2016, vol. 30, 2P113in1-2P113in1, 2016

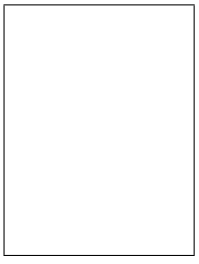
Shota Akiyoshi received a B.E. degree from Kyushu Institute of Technology, Iizuka, Japan in 2021. Since April 2021, he has been a Master course student at Kyushu Institute of Technology. His recent research interests are a network and computational resource allocation method.



Yuzo Taenaka received a D.E. degree in information science from Nara Institute of Science and Technology (NAIST), Japan, in 2010. He was an Assistant Professor with the University of Tokyo, Japan. He has been an Associate Professor with the Laboratory for Cyber Resilience, NAIST, since April 2018. His research interests include information networks, cybersecurity, distributed systems, and software defined technology.



Kazuya Tsukamoto received a DE degree in computer science from the Kyushu Institute of Technology (Kyutech), Japan, in 2006. From April 2006 to March 2007, he was a Japan Society for the Promotion of Science (JSPS) research fellow at Kyutech. In 2007, he was an assistant professor in the Department of Computer Science and Electronics, Kyutech, and then was an associate professor in the same department in 2013. He has been a professor in the same department since January 2022. His research interests include performance evaluation of computer networks and wireless networks. He is a member of the ACM, IPSJ, IEICE, and IEEE.



Myung Lee received BS and MS degrees from Seoul National University, and a Ph.D from Columbia University. He is currently a professor with the Dept. of Electrical and Computer Engineering, City University of New York. His current research activities include Secure V2X communications, Edge Cloud resource management, IoT, machine learning, and stochastic computing applications to intrusion detection. He publishes extensively in these areas and hold over 25 U.S. and international patents. His research group developed the first NS-2 simulator for IEEE 802.15.4, a standard NS-2 distribution widely used for wireless sensor network research. Dr. Lee co-received three best paper awards. He actively contributed to international standard organizations IEEE (TG chairs for 802.15.5 WPAN Mesh and 802.15.8 PAC) and ZigBee. He is a past president of the KSEA.