Matching based content discovery method on Geo-Centric Information Platform

Kaoru Nagashima, Yuzo Taenaka, Akira Nagata, Hitomi Tamura, Kazuya Tsukamoto, Myung Lee

Abstract We have proposed a concept of new information platform, Geo-Centric information platform (GCIP), that enables IoT data fusion based on geolocation. GCIP produces new and dynamic contents by combining cross-domain data in each geographic area and provides them to users. In this environment, it is difficult to find appropriate contents requested by a user because the user cannot recognize what contents are created in each area beforehand. In this paper, we propose a content discovery method for GCIP. This method evaluates the relevancy between topics specified in user requests and topics representing IoT data used for creating contents, called matching, and presents the candidates for the desired contents based on the relevancy. Simulation results showed that appropriate contents can reliably be discovered in response to user's request.

Kaoru Nagashima
Kyushu Institute of Technology , Japan e-mail: nagashima@infonet.cse.kyutech.ac.jp

Yuzo Taenaka
Nara Institute of Science and Technology, Japan e-mail: yuzo@is.naist.jp

Akira Nagata
iD Corporation, Japan e-mail: a-nagata@intelligent-design.co.jp

Hitomi Tamura
Fukuoka Institute of Technology, Japan e-mail: h-tamura@fit.ac.jp

Kazuya Tsukamoto
Kyushu Institute of Technology, Japan e-mail: tsukamoto@cse.kyutech.ac.jp

Myung Lee
CUNY, City College, USA e-mail: mlee@ccny.cuny.edu

1 Introduction

The cross-domain (horizontal-domain) IoT data fusion attracts much attention. To create contents based on local cross-domain IoT data, linking geolocation to IoT data is essential. Even though, as each of network is managed without the consideration of the physical location, the feasibility of the cross-domain data fusion is quite low in the practical environment.

So far, we have proposed Geo-Centric Information Platform (GCIP) [1] that conducts a geolocation-oriented IoT data collection/processing. Although each IoT device is supposed to dedicate to a particular service, GCIP collects data created at the physical proximity and produces contents from them as for secondary use by nearby edge servers. GCIP introduces two kinds of edge servers, a data store (DS) server and data fusion (DF) servers, for these processing. A DS server, which is assumed to be arranged for a particular area and managed by local organization like mall-owner or municipality, stores the collected IoT data. A DF server(s) deployed by contents service providers produces new and dynamic contents using cross-domain IoT data stored in the DS server. In this paper, we define contents created by DF server(s) as spatio-temporal content (STC).

As IoT data like sensor data dynamically changes their number, sensing preciseness, existence, etc. by various reasons, such as movement and sleep cycle of IoT devices, DF server may accordingly produce different STCs depending on the observable IoT data at that point. Therefore, users hardly recognize variable STCs created in a timely manner. Moreover, as the number of DF servers (contents providers) may differ between geographical areas, it is quite difficult to identify every DF servers as well as recognize what STC is created in which DF server. Therefore, GCIP needs to provide users with a mechanism that finds appropriate STCs they are interested in.

In this paper, we propose a content discovery method that finds the candidates for STCs the user is requesting. Specifically, in the method, a DS server evaluates the relevancy between topics specified in user requests and topics representing IoT data used for creating STCs, and then selects a DF server having the highest potential to have appropriate STCs. The designated DF server searches for appropriates STCs from its own STC database and presents a list of STC candidates to users. We evaluate the effectiveness of our proposed discovery method on GCIP through simulation.

The rest of this paper is organized as follows, we first review the existing studies in Section 2. Then, we describe the requirements for STC management on the GCIP in Section 3. Proposed method is described in Section 4 and simulation environment and simulation results are explained in Section 5. Finally, the conclusion is provided in Section 6.

2 Related Work

In this section, we review the existing studies focusing on geo-location based cross-domain data fusion and IoT contents discovery. Reference [2] summarizes the existing studies focusing on smart city applications, which use IoT data for users in some city. Various sensors are deployed on the platform in which IoT data obtained from these sensors are used for a specific application only. On the other hand, as our proposed GCIP is designed for cross-domain IoT data fusion, variety of purposes are aimed to be supported even in a smart city constructed by multiple service providers. Therefore, our study flexibly produces contents without any limitations on the purpose of sensors and networks.

References [3] and [4] show examples for smart city application using cloud platform. In these papers, all IoT data are sent to the cloud, processed, and then delivered to the user from the cloud. On the other hand, GCIP processes the collected IoT data at edge servers; there is no need to carry them to the cloud, so that people can benefit from local use in terms of not only a few load on the network but also a low latency.

Reference [5] summarizes the existing studies focusing on the contents search method from various perspectives (e.g., event-based [6], metadata-based [7], and location-based [8]). Although none of existing studies tries to find the contents dynamically created from the collected IoT data, the proposed method can achieve it.

Information Centric Network (ICN) is a promising concept that brings us efficient content search and distribution. As the ICN operated in a content-based, not an IP-based [9], users can directly search for a content by using content name without knowing the location of the content. However, since the contents are created from cross-domain IoT data observable at that time, the user cannot know their name at the retrieval timing in the ICN, which makes us extremely difficult to search for the contents. As a result, we can say that the proposed contents discovery method is novel.

3 STC Management on the GCIP

This section briefly describes the concept of STCs management on the GCIP [1], which comprises of three steps: IoT data collection, STC production, and STC discovery, as shown in Fig. 1.

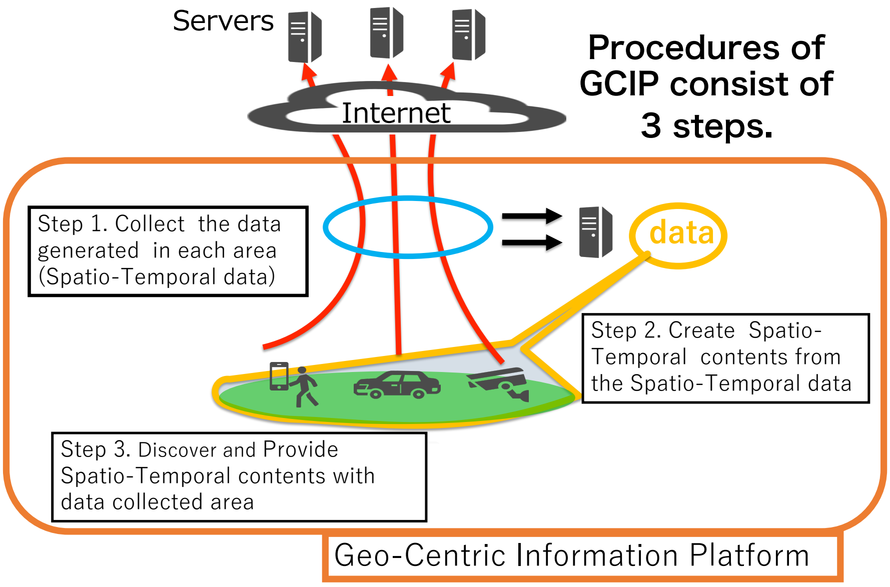
Servers
Internet
Procedures of GCIP consist of 3 steps.
Step 1. Collect the data generated in each area (Spatio-Temporal data)
data
Step 2. Create Spatio-Temporal contents from the Spatio-Temporal data
Step 3. Discover and Provide Spatio-Temporal contents with data collected area
Geo-Centric Information Platform

Fig. 1 Conceptual design of GCIP

3.1 Step1: IoT data collection

As shown in Fig. 2, GCIP divides geographical area into hierarchical meshes based on latitude and longitude lines, and each of which has a unique mesh code. There is a router, called mesh router, to be responsible for the routing function in the corresponding mesh. Since we embedded mesh codes, which uniquely identify every meshes, into IPv6 address format [10], every packets following the GCIP rules can be processed (e.g. routing) with the consideration of geolocation specified in IPv6 address even during the transmission of traditional Internet.

3.2 Step2: STC production

Since a DS server is responsible for storing data in its belonging mesh and DF servers are responsible for creating STCs, deployment of DS server and DF server(s) are necessary for creating STCs. In our method, Publish/Subscribe model is applied for the communication among these two sorts of servers. The mesh router publishes the collected IoT data to the DS server. Each of the DF servers sends subscribe requests to the DS server for data represented by the desired topics. The DF server produces a STC by using those data.

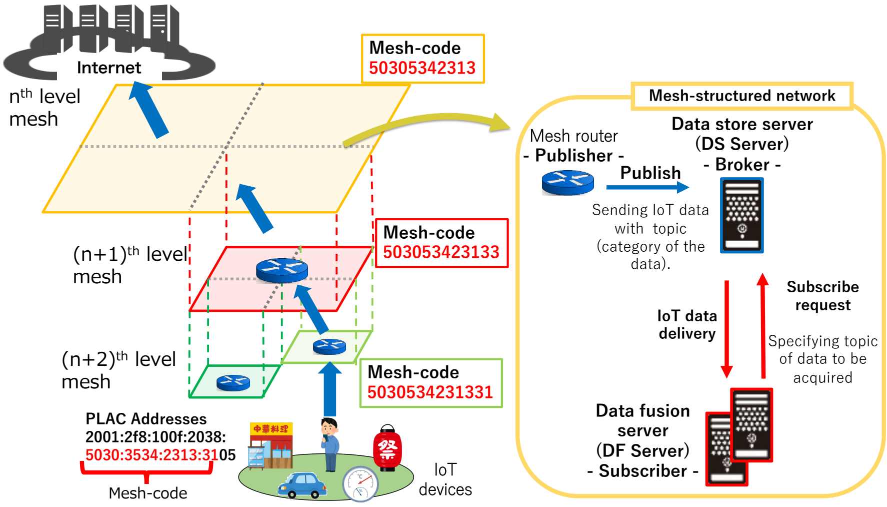

Fig. 2 STC generation image in GCIP

In this paper, we assume that one STC will be created by data collected by a single subscribe request including multiple topics.

3.3 Step3: STC discovery

The data collected from diverse IoT devices will change spatio-temporally in terms of their amount and generation interval. As a result, STCs created by the DF servers have spatio-temporal characteristics. In such a case, users who want to get a STC on the GCIP hardly discover which DF server has what STCs. Therefore, we need to consider a new STC discovery method that can satisfy the following requirements: (1) support for geolocation-aware discovery, (2) support for fuzzy search and (3) support for exploration of new findings. Note that, as we already proposed a geolocation-oriented communication, we already satisfy the requirement (1).

4 Proposed method

Since we assume that each of DF servers sends a subscribe message including desired topics to the DS server for creating a STC, the DS server can grasp the statistics on the number of topics included in all of subscription messages. On the other hand, users need to send a request message with desired topics to the DS server because they cannot directly identify which DF server has what STCs. Therefore, we try to evaluate the relevance matching between the user request and statistical information of subscriptions at the DS server. Through this evaluation, the DS server identifies a DF server that has the most appropriate STCs created by relevant topics with the user's request, and forwards the user request message to have the DF server search for appropriate STCs in response to the user's request.

In the proposed method, a user is supposed to specify mesh-code, topic, and priority of each topic in a request message. Regarding topics, a request message contains up to three topics, each of which is allocated with the variable weight. The weight of each topic is different depending on the user preferences, but the total must be 100. For example, if the three topics are called T_1, T_2, and T_3, a user specifies the weight 70 for T_1, 20 for T_2, and 10 for T_3. In this way, even if a user has no knowledge about STCs, the user sends a request including topics with their weight (priority), not any particular identifier of STCs, thereby achieving requirement (2).

As the DS server obtains the statistics on subscription messages transmitted from every DF servers, we use the statistics for STC discovery. The DS server calculates a subscription probability $P_{i,j}$ of topic j for each DF server i as follows,

$$P_{i,j} = \frac{S_{i,j}}{S_{i,\mathrm{all}}} \tag{1}$$

where $S_{i,j}$ is the number of topic j subscribed by the DF server i and $S_{i,\mathrm{all}}$ is the total number of topics subscribed by the DF server i. From the assumption of one STC from one subscription including multiple topics, the DF server with the highest $P_{i,j}$ has the largest number of STCs partly created by the topic j.

To find appropriate STCs based on both the user request and the subscriptions statistics, we evaluate the relevance between them. Although a user request includes up to three topics, there may not be STCs created by the exact same topics. We then search for STCs created by not only exact same topics with user request but also relevant STCs created by different combinations of topics, thereby providing unconscious users' interest. Specifically, as we have 10 sets of possible combinations out of three topics in total, i.e., ${}_3C_1$ + ${}_3C_2$ + 1= 10, we choose several combinations, a set of partial combinations C, for searching. That is once, the DS server receives a user request, it calculates total weight for every possible combinations of topics based on weight specified in the request, and then only selects combinations whose total weight exceeds the predetermined threshold α. That is, these combinations are the partial combinations C.

To identify an appropriate DF server, the DS server calculates the expected value, E_i, of the number of STCs satisfying the threshold for each DF server i.

$$E_i = \sum_{c \in C} (G_i(c) \times N_i) \tag{2}$$

where C is a set of partial combinations of user-specified topics, whose total weight exceeds the threshold. $G_i(c)$ is defined as $\prod P_{i,j}$ for all topics contained in c. N_i is the total number of subscription messages received by the DF server i. This number essentially means the number of STCs created by the server.

Since a DF server with the highest value of E_i is most likely to have the large number of STCs requested by the user, the DS server forwards the request to the DF server. After that, the DF server searches for STCs created by several combinations of topics, whose total weight exceeds α, in its own STC database. In this way, as several STCs may be discovered based on not only topics specified in the user request but also other topics, a list of such candidate STCs can be presented to the user as a response to the request. From these consideration, we can remark that our proposed method successfully satisfies search requirement (3).

Finally, we describe an example where a user retrieves STC. A user, one of us, may want to know a temperature and humidity to choose clothes before visiting anywhere. The user sends a request with the topics of temperature and humidity regarding the location where the user will go. As a result, the user can get not only requested STC, i.e., temperature and humidity but also several STCs related to those topics if the user discovers STCs by the proposed method. In this case, we may be able to get both the discomfort index [1] made by temperature and humidity and sensible temperature made by temperature, humidity, and wind speed. In this context, the proposed method brings user's unconscious interests.

5 Evaluation for STC discovery

5.1 Simulation environment

We conduct simulation to evaluate our proposed method. Fig. 3 shows the network topology we employed in this simulation. Each DF server individually produces STCs with various combinations of randomly selected topics, which are 2 to 5 topics, as shown in Table 1.

In order to ensure the feasibility of STCs production process, we additionally employ the subscription possibility parameter, called subscription bias, which is the popularity of topics used for STC production on a DF server. That is, if the subscription bias of a certain topic is set to 1, the DF server must include the topic in all of the subscriptions sent to the DS server. This case is called subscription bias 1. If it is set to 0.5, every DF server uses a particular topic in 50 percent possibility to produce STCs. We define this condition as subscription bias 0.5. Note, each DF server randomly sets the number of topics up to maximum of 5 in addition to a topic determined by the subscription bias.

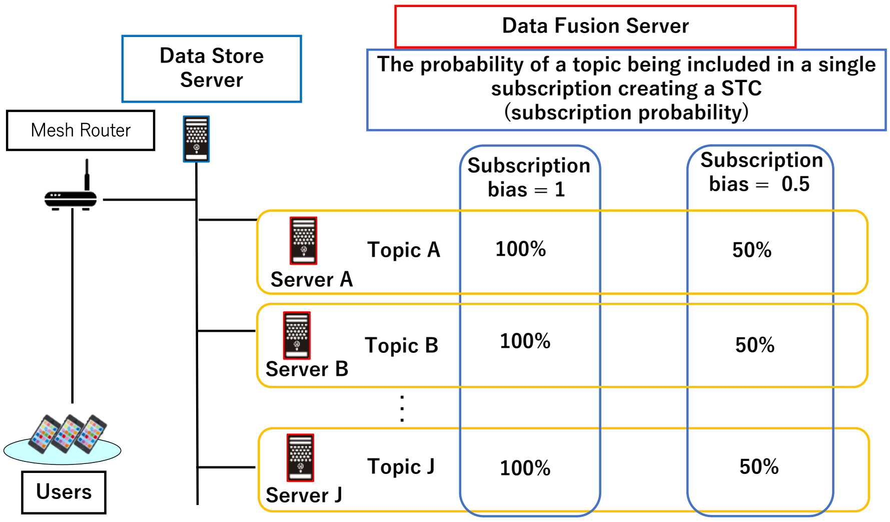

Fig. 3 Simulation environment for performance evaluation

Table 1 Simulation parameters

Parameters	Value
The number of DF servers	10
The number of STCs created on each DF server	1000
The number of topics	10
The number of topics used for each STC	2∼5
Threshold of the total weight α	80

Table 2 User request bias settings

	Weight of T_1	Weight of T_2	Weight of T_3
Request bias 1 (imbalanced)	80	15	5
Request bias 2 (relatively balanced)	35	35	30

We assume that up to three topics characterizing the desired STCs are included in a user request. Since the topics have different weight, we introduce the user request bias, in addition to the subscription bias, as shown in Table 2. We use two patterns for user request bias: request bias 1 (imbalanced) and request bias 2 (relatively balanced). In each simulation, we send 10 user requests in one round and then conduct the experiment for 10 rounds.

5.2 Comparative methods and performance measures

We use the following two comparative methods to show the effectiveness of the proposed method.

Comparative method 1: This method searches for STCs whose configuration topics completely match topics included in the user request.

Comparative method 2: This method searches for STCs from a DF server that is expected to hold the largest number of STCs configured from T_1 topic.

We introduce the new performance measure, called unconscious contents ratio (UCR), to evaluate the amount of STCs that offer more topics than those of users' request. As the proposed method enables to provide STCs made by relevant topics, the user has a potential to retrieve unconscious but interested STCs. That is why the evaluation of UCR is also important. The UCR is calculated by Equation (3).

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

where C_{total} shows the total number of retrieved STCs and $C_{complete\_match}$ shows the number of STCs created from specified topics only.

5.3 Simulation results

We describe the number of retrieved STCs and the UCR through the simulation. Table 3 and 4 show the results in case of subscription bias with 1 and 0.5, respectively. Since the comparative method does not consider the weight of the specified topics, the results are similar, irrespective of the change in the request bias.

In contrast, the proposed method retrieves the relatively larger number of STCs that involves one and/or two additional topics not limited to specified by the user. They are referred to as complete match+1, complete match+2, respectively. The overall number of STCs in case of subscription bias = 0.5 is lower than that of subscription bias = 1. However, regardless of subscription bias, there is little difference in the number of complete match STCs on the comparative method 1, 2, and the proposed method. Since all the methods including the proposed method are able to estimate a DF server holding the appropriate STCs, there is no difference between the selected DF server among all methods.

In contrast, the proposed method can retrieve 9.4 to 9.9 times larger number of STCs, regardless of the subscription bias because the proposed method obtains the STCs involving complete match+1 and complete match+2.

Table 3 The retrieved number of STCs (Subscription bias = 1)

	Comparison method 1	Comparison method 2	Proposed method		
	Complete match	Complete match	Complete match	Complete match+1	Complete match+2
Request bias 1	7.65	7.24	7.24	20.63	41.67
Request bias 2	7.65	7.24	7.65	22.37	43.94

Table 4 The retrieved number of STCs (Subscription bias = 0.5)

	Comparison method 1	Comparison method 2	Proposed method		
	Complete match	Complete match	Complete match	Complete match+1	Complete match+2
Request bias 1	3.71	3.58	3.58	10.04	21.5
Request bias 2	3.71	3.58	3.71	11.04	22.2

Table 5 UCR in case of subscription bias=1

	UCR	95% confidence interval
Request bias 1	89.6	0.454
Request bias 2	89.6	0.700

Table 6 UCR in case of subscription bias=0.5

	UCR	95% confidence interval
Request bias 1	90.0	1.358
Request bias 2	89.9	0.756

Finally, we assess the UCR. Table 5 shows how the UCR is varied with the change in the user request bias, in case of the subscription bias = 1. Table 6 shows how the UCR is varied with the change in the user request bias, in case of the subscription bias = 0.5. From these results, we demonstrate that the UCR does not change, irrespective of the changes in not only subscription bias but also the request bias. Therefore, the proposed method provides not only the desired STCs but also the unconscious STCs, even under any environments of topic subscriptions and user requests.

6 Conclusion

We have proposed the Geo-Centric Information Platform (GCIP) that can provide Spatio-temoral Contents (STC) for specific area. However, from the view point of user, since the STC is dynamically created on the GCIP, a method of STC discovery is necessary. In this paper, we proposed a method that matches topics specified in user requests and topics representing IoT data used for creating STCs. Through simulation, we showed that the user can retrieve the relatively larger number of STCs including both required and non conscious (but beneficial) topics.

We showed that the proposed method can flexibly perform STCs search in accordance with user requirements. The user is possible to retrieve appropriate STCs even if the user does not know the STC information beforehand. Although the current proposed method tries to discover appropriate STCs only in a mesh, we have to take into account the cases where there is no appropriate STC in the specified mesh. Therefore, we will extend the proposed STC discovery method to adaptively change the size of geolocation area for achieving the reliable STC discovery.

Acknowledgements This work was supported by JSPS KAKENHI Grant Number JP18H03234, NICT Grant Number 19304, USA Grant number 1818884, and 1827923.

References

1. K. Nagashima, Y. Taenaka, A. Nagata, K. Nakamura, H. Tamura, K. Tsukamoto, "Experimental Evaluation of Publish/Subscribe-based Spatio-Temporal Contents Management on Geo-Centric Information Platform, " Advances in Networked-based Information Systems, Vol. 1036, pp. 396-405, August 2019.
2. B.P.L. Lau, S.H. Marakkalage, Y. Zhou, N.U. Hassan, C. Yuen, M. Zhang, U. X. Tan, "A survey of data fusion in smart city applications, " Information Fusion, Vol. 52, pp. 357-374, December 2019.
3. S. Consoli, D. Reforgiato, M. Mongiovi, V. Presutti, G. Cataldi, W. Patatu, "An urban fault reporting and management platform for smart cities, " WWW '15 Companion: Proceedings of the 24th International Conference on World Wide Web, pp 535-540, May 2015.
4. F.Ahmed, Y.E. Hawas, "An integrated real-time traffic signal system for transit signal priority, incident detection and congestion management, " Transportation Research Part C: Emerging Technologies, Vol. 60, pp 52-76, November 2015.
5. S. Pattar, R. Buyya, K.R. Venugopal, S.S. Iyengar, L.M. Patnaik, "Searching for the IoT Resources:Fundamentals, Requirements, Comprehensive Review, and Future Directions, " IEEE Communications Surveys & Tutorials, Vol. 20, pp. 2101-2132, April 2018.
6. A. Pintus, D. Carboni, A. Piras, "Paraimpu: A platform for a social Web of Things, " in Proc. 21st Int. Conf. Companion World Wide Web (WWW Companion), pp. 401-404, April 2012.
7. S. Mayer, D. Guinard, "An extensible discovery service for smart things, " WoT'11: Second International Workshop on the Web of Things, June, pp. 1-6, 2011.
8. S. Mayer, D. Guinard, V. Trifa, "Searching in a web-based infrastructure for smart things, " 2012 3rd IEEE International Conference on the Internet of Things, pp. 119-126, October 2012.
9. G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, G. Polyzos, "A survey of information-centric networking research," IEEE Communications Surveys Tutorials, Vol. 16, pp 1024-1049, July 2013.
10. Hitomi Tamura, "Program for determining ip address on the basis of positional information, device and method, " JP Patent 6074829, 2017-01-20.