Loose matching approach considering the time constraint for spatio-temporal content discovery

Shota Akiyoshi, Yuzo Taenaka, Kazuya Tsukamoto, Myung Lee

Abstract Cross-domain data fusion is becoming a key driver in the growth of numerous and diverse applications in the IoT era. We have proposed the concept of a new information platform, the Geo-Centric Information Platform (GCIP), that enables IoT data fusion based on geolocation. The GCIP dynamically produces spatio-temporal content (STC) by combining cross-domain data in each geographic area and then provides the STC to users. In this environment, it is difficult to find some particular STC requested by a user because the user cannot determine which STC is created in each area beforehand. Although, in order to address this difficulty, we proposed a content discovery method for GCIP in the previous study, the temporal property of STC was not taken into account, despite the fact that the available (effective) period of each of STC is limited. In the present paper, we propose a new loose matching approach considering the time constraint for STC discovery. Simulation results showed that the proposed method successfully discovered appropriate STC in response to a user request.

Shota Akiyoshi
Kyushu Institute of Technology, Japan e-mail: akiyoshi.shota190@mail.kyutech.jp

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

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

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

1 Introduction

With the development of IoT technologies, the collaboration of cross-domain data obtained by sensing a wide variety of things has attracted attention [1]. We proposed the Geo-Centric Information Platform (GCIP) [2], which collects, processes, and distributes IoT data (i.e., realizes IoT data fusion) in a geolocation-aware manner. As shown in Fig. 1, the GCIP divides the space into a hierarchical mesh structure based on longitude and latitude, configures a network corresponding to the mesh structure, and generates content from the IoT data collected in each mesh. Two types of servers, a data store server (DS server) and a data fusion server (DF server), are deployed in a mesh. The DS server collects all IoT data in the geospatial range corresponding to the mesh, and the DF server uses this IoT data and generates spatio-temporal content (STC), which are local content for the region.

The data properties are various in terms of collection interval, data volume, and data presence, depending on the type and/or movement of IoT devices. The DF server generates STCs dynamically according to the data present at the time of STC creation. That is, the generated STC depends on the IoT data being collected at that time. Therefore, it is difficult for users to find particular STC because they cannot know the STC generated by the DF server in advance.

The previous study [3] proposed an STC discovery method that enabled users to find appropriate STCs by matching the statistics of IoT data usage of each DF server with user requests. Although this method can select the DF server having the largest amount of STC, there is a problem in which the DF server which does not have the largest amount of STC is never selected, even if the DF server has appropriate STC for the user request. In addition, STCs has an available period, but that is not taken into account.

In the present paper, we propose an extended STC discovery method that considers the available period of STC. The proposed method enables the discovery of servers that have a large amount of STC, which is appropriate for user requests and also remains available by exploiting cosine similarity.

The remainder of the present paper is organized as follows. Section 2 introduces related research on content retrieval, and Section 3 describes our previous research. Section 4 describes the proposed method, and Section 5 explains the comparison method, valuation metrics, and simulation. Section 6 presents a summary of our paper.

2 Related Research

In this section, we review existing content retrieval methods. Reference [4] summarizes existing studies that focus on content retrieval (location-based [5], metadata-based [6], and event-based [7]). These methods use the elements

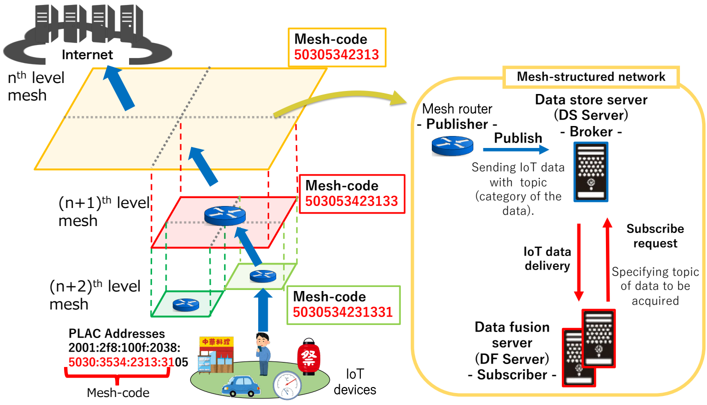


Fig. 1 Assumed environment for GCIP

of time, location, and content as different search metrics. Information-centric networking (ICN) is a promising concept for efficient content retrieval and distribution. Since ICN operates on a content basis rather than an IP basis, users can directly search for content using the content name without knowing the location of the content [8]. However, in the case of cross-domain data fusion, the several content are created in response to IoT data collected at that time, and the user cannot know the name of the content at the time of search in ICN. This makes it very difficult to search for the content. In the present paper, we use topics that are used to compose content as a metric for content search.

3 GCIP: Geo-Centric Information Platform

The GCIP can arrange a transmission route based on physical location because it uses a network address with a unique ID called the mesh ID, which is a hierarchically defined number that depends on the mesh structure as shown in Fig. 1. This enables location-aware search only by designating a particular mesh ID in a request packet of content search. In the present study, we focus on STC retrieval by matching the statistics of IoT data usage of the DF server with the user request after a search packet reaches the designated mesh.

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 the region. In the present study, we assume that there is one DS server and several DF servers in a mesh of the GCIP.

In order to make STC generation occur asynchronously, we use Publish/Subscribe (Pub/Sub) communication (Fig. 1). The mesh router is 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 DS server with a topic indicating the type of data. Upon 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. At this time, the available period of each STC is set by the DF server. In the present study, we assume that one STC is generated by collected data for one subscription request and that the DS server cannot understand whether a subscription consisting of the same topics is for the same STC or for a different STC.

4 Matching-based STC Discovery

4.1 Conceptual design of the matching search method

Since each DF server generating STC is managed by a different operator and the type and/or timing of content generated by each DF server vary, users cannot know what type of STC is being generated or when and where this generation occurs and thus cannot directly request a search of any DF server. In this circumstance, since for a user to specify the name of content or explicit keywords is also difficult, 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 mesh at the time of STC generation and thus can use its statistical information.

The following is an overview of the matching-based search. The user first sends a search packet to the DS server, which will be the anchor point in each mesh because the DS server has all the data in each mesh and has subscription statistics. When the DS server receives the search packet, 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 packet to the DF server, and the DF server sends the user STC(s) 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 of the proposed method.

4.2 Previous search method and the 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 $G_i(c) = \prod P_{i,j}$. We use these definitions, and the expected value of the amount of STC on DF server i can be expressed as $E_i = \sum_{c \in C}(G_i(c) \times N_i)$.

Although this method can select a DF server having the largest amount of STC, it is two problems in that the DF server that does not have the largest amount of STC can never be selected, even if that server has appropriate STC for the user request. In addition, since all STC have an available period, the STC should be taken into account in the search procedure. Without this consideration, a selected DF server might have only old (not useful) STC.

4.3 Proposed method

In the proposed method, a user specifies three types of information for STC discovery: the location information of the target area, search keyword of the desired content, and ambiguity level. The search keywords are translated (or decomposed) to topics by using any intention extraction technique. 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. The ambiguity level, a number from 0 (strictly same) to 100 (allowing anything), means that the degree the user allows the search results to differ from the specified keywords. 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 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 of a user request, and in which the remaining available period is long. 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 previous 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 available period of all appropriate STC, respectively. The score of 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 of 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. In this way, servers with a large amount of only old STC, or servers with a small amount of STC but a very long available period for one piece of STC, are not selected, and a server with a large amount of STC and STC with a long available period for the entire STC can be determined as the optimal server.

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

Figure 2 shows the matching procedure of the proposed method. In order to identify an optimal DF server, a DS server estimates an optimal DF server by matching the subscription statistics of DF servers and 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 of priority value, this can be treated such that the user expects STC composed in part from topics with the ratio of priority value. From this similar context, a DS server that can have 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 on the subscription of the DF server and user request. The DS server keeps the combination of topics subscribed by each

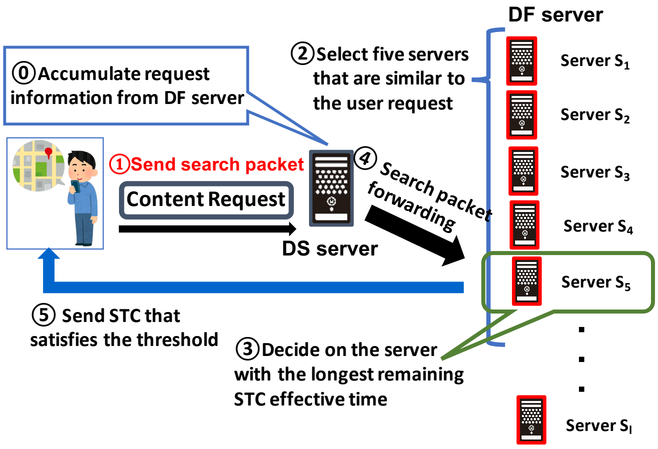

DF server
②Select five servers that are similar to the user request
⓪Accumulate request information from DF server
①Send search packet
Content Request
④Search packet forwarding
DS server
Server S_1
Server S_2
Server S_3
Server S_4
Server S_5
Server S_l
⑤Send STC that satisfies the threshold
③Decide on the server with the longest remaining STC effective time

Fig. 2 Matching procedure of the proposed method

DF server, the last subscription time of 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 subscription information of the DF server and user request:

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

Next, we calculate the ratio of the combination c_m to the subscriptions of one DF server. We set the ambiguity 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 ambiguity level, α, which could be that of only one topic, even if a user request includes several topics. This is to make a search result involving related information. Define the weight for each element in c_{m_α} as $w_{c_m} = n_{c_m}/n_{\text{sum}}$. Define the weight w_j of topic j as the sum of the w_{c_m} 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. 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 in Eq. 1 is estimated to be the optimal DF server. However, from the simulation result, it is clear that the server with the highest cosine similarity is not necessarily the optimal DF server. We investigated the distribution of optimal DF servers when they are arranged in the order of the cosine similarity value, and the range of possible values of the cosine similarity. We investigate the order of the cosine similarity of the optimal DF servers when CS_l calculated by Eq. 1 is arranged in descending order. From the results, it can be seen that about 90% of the optimal DF servers are contained in the top five in terms of cosine similarity, and, therefore, the servers that are within the top five servers in terms of cosine similarity and satisfy the thresholds (MIN = 0.82, MAX = 0.97) are considered as candidate search results.

4.3.3 Stage 2: Optimal DF server selection

The next step is to select one of the five candidates chosen in Section 4.3.

We use the available period to select a candidate. The DF server with the largest median STC available period is defined as the server with the longest available period. This definition eliminates the possibility of selecting a server having old STC. However, since the DS server cannot know the available period of each STC, it uses the value of the Poisson distribution p_{c_m} of the mean available period λ to estimate the distribution. Specifically, the remaining available period of the combination c_m is calculated as e_{c_m} in Eq. 2. The arrival time of the search packet of the user is set to t_{now}.

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

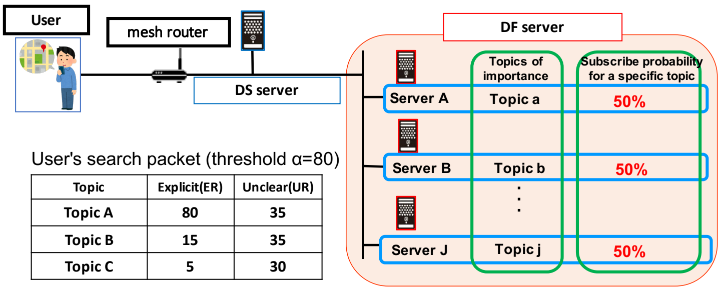
Let U_{S_l} be the median of the e_{c_m} aggregate of DF servers S_l. Among all the DF servers in the mesh, the server with the largest U_{S_l} has the highest probability of being the optimal DF server, so the DS server forwards the search packet of the user and performs the search.

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.

5 Performance Evaluation

5.1 Simulation environment

We use the simulation environment focusing on a mesh, as shown in Fig. 3. There are 10 DF servers, and each DF server is biased to request many



Topic	Explicit(ER)	Unclear(UR)
Topic A	80	35
Topic B	15	35
Topic C	5	30

Fig. 3 Simulation topology

Table 1 Simulation parameters

Number of servers	10 units
Number of content generated per unit of time	100[piece/unit]
Topic Type	10[types]
Number of topics linked	2∼5
STC mean period to available	10,30,60[minutes]
Threshold α	80
Mean period to available λ	10,30,60[minutes]

subscriptions to a particular topic. The probability of obtaining a specific topic at the time of the subscribe request is set to 50% (Subscription bias 0.5: hereafter SB = 0.5) and 100% (SB = 1). In addition, we assume that each DF server requires several different topics. The number of topics constituting STC is set randomly. We search by an unclear request (UR) in which there is little difference in the importance of each topic. In the simulation, STC is generated for 10 minutes, and then a search packet is sent for evaluation. In this simulation, we set the parameters as shown in Table 1. The user sends an UR to the DS server 1,000 times to evaluate the performance of the search.

5.2 Evaluation index

We use three evaluation indexes: 1) the estimation accuracy, which is the probability that each method is estimated a DF server of each rank; 2) the amount of appropriate STC, which matches the user request, from the estimated server; and 3) the distribution of the remaining available period of the obtained STC (Eq. 2). Note that, we call the condition in which the obtained STC is composed of exact the same topics as included in a user request as cp. Furthermore, we may say cp+1 when the amount of STC obtained that contains one topic other than the user request, and cp+2 is the amount of STC obtained that contains two topics.

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, which is referred to as expected value basis (EV). Comparison method 2 uses only the cosine similarity to select the appropriate DF server, which is referred to as Cosine similarity basis (CS). Since the difference with the proposed method does not take the remaining available period into account, we can evaluate its effectiveness on search results. 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

Figure 4 shows the ratio of the identified DF server on the experiments of two environments in which user request (UR) and subscription bias (SB) 1.0/0.5 are combined. This figure show the data for the proposed method, Expected Value basis (EV, comparison method 1), and Cosine similarity (CS, comparison method 2), respectively. The order of the identified DF server, denoted as 1st, 2nd, and 3rd, is from the order of score, which is used for the definition of the optimal DF server in Section 4.3.1. Here, 1st indicates a method found the optimal DF server. For the case in which both URs and SBs are highly biased (SB=1, UR), the proposed method is able to estimate the DF servers with more than the top three DF servers with 92% accuracy. In the environment of UR and SB=0.5, the proposed method can estimate more than the top three DF servers with 81% accuracy. Next, Table 2 shows the average value of the amount of STC acquired by users in each SB. Figure 5 shows the remaining available period of the STC acquired by users. We show the results of the proposed method and the CS based method, which collects a large amount of STC.

Figure 4 shows that the proposed method has the highest accuracy in estimating the optimal DF server in all environments, with 47% accuracy even for SB = 0.5 and UR. The accuracy of the cosine similarity-based method has the next highest result, with SB = 0.5, and 20% for UR.

EV has an estimation accuracy of less than 1% in all environments. EV is a method for selecting the server with the largest amount of STC that satisfies the user requirements and does not take into account the available period of STC. As a result, EV fails to select the best DF server by the fifth server, and the worst server among the 10 DF servers is selected most often. CS is a significant improvement over EV and is able to select the optimal DF server. Since EV does not take cp+1 and cp+2 into account, it is not possible to estimate the DF server that maximizes the acquisition STC including cp+1 and cp+2. Since the time characteristics are not taken into account, the number of times that the optimal DF server was estimated was less than that for the proposed method in all environments.

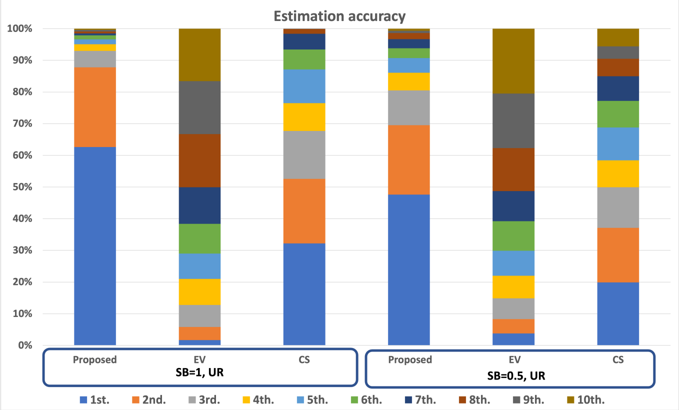


Fig. 4 Estimation accuracy (proposed method)

Table 2 Number of STC acquisitions

	UR(SB=1)			UR(SB=0.5)		
	cp	cp+1	cp+2	cp	cp+1	cp+2
Proposed method	4	15	29	3	8	14
Expected value basis (EV)	0	3	11	1	4	13
Cosine similarity based (CS)	8	20	32	5	10	12

The difference in median available period between the proposed method and the STC obtained by CS was approximately 4 minutes for SB = 1 and UR with small bias and approximately 5 minutes for SB = 0.5 and UR. The proposed method can provide STC in SB = 0.5 and UR environments with a longer available period than the methods with similarity.

In all methods, we can obtain STC that include topics not yet specified by the user, which can give the user new insights. 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.

6 Conclusion

In the GCIP, users cannot know when and where any STC is generated, nor can they directly request any DF server to search for STC. Therefore, we proposed a matching approach for STC that satisfies the user request by focusing on the similarity between the subscription statistics of DF servers and the user request and the available period of STC. The simulation results showed that the user can obtain fresh STC. In the future, we intend to estimate the amount of STC generated from the transmission interval of

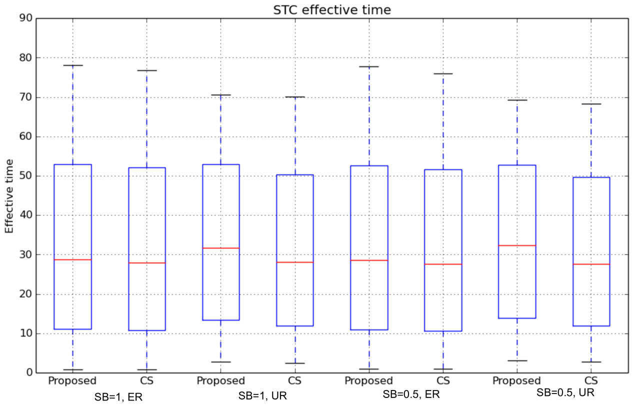


Fig. 5 Remaining available period of acquired STC

the same subscription and to study methods by which to improve the search accuracy.

Acknowledgements The present study was supported by JSPS KAKENHI Grant Number JP21H03430, by NICT Grant Number 19304, and by USA Grant numbers 1818884, and 1827923.

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, June 2015.
2. 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.
3. K. Nagashima, et al. "Matching based content discovery method on Geo-Centric Information Platform, " INCoS 2020, vol 1263, pp 470 – 479 Sept 2020.
4. 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).
5. 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, October 2012
6. Mayer, S., Guinard, D.: An extensible discovery service for smart things. In: WoT 2011: Second International Workshop on the Web of Things, June, pp. 1-6 (2011)
7. Pintus, A., Carboni, D., Piras, A.: 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
8. G. Xylomenos, et al.: A survey of information-centric networking research. IEEE Commun. Surv. Tutorials vol 16, pp. 1024-1049 (2013)