Research Article

Two-Tier VoI Prioritization System on Requirement-Based Data Streaming toward IoT

Sunyanan Choochotkaew, Hirozumi Yamaguchi, and Teruo Higashino

Graduate School of Information Science and Technology, Osaka University, Osaka, Japan

Correspondence should be addressed to Sunyanan Choochotkaew; sunya-ch@ist.osaka-u.ac.jp

Received 5 March 2017; Revised 27 June 2017; Accepted 2 July 2017; Published 20 August 2017

Academic Editor: Laurence T. Yang

Copyright © 2017 Sunyanan Choochotkaew et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Toward the world of Internet of Things, people utilize knowledge from sensor streams in various kinds of smart applications. The number of sensing devices is rapidly increasing along with the amount of sensing data. Consequently, a bottleneck problem at the local gateway has attracted high concern. An example scenario is smart elderly houses in rural areas where each house installs thousands of sensors and all connect to resource-limited and unstable 2G/3G networks. The bottleneck state can incur unacceptable latency and loss of significant data due to the limited waiting-queue. Orthogonally to the existing solutions, we propose a two-tier prioritization system to enhance information quality, indicated by *VoI*, at the local gateway. The proposed system has been designed to support several requirements with several conflicting criteria over shared sensing streams. Our approach adopts Multicriteria Decision Analysis technique to merge requirements and to assess the *VoI*. We introduce the framework that can reduce the computational cost by precalculation. Through a case study of building management systems, we have shown that our merge algorithm can provide 0.995 cosine-similarity for representing all user requirements and the evaluation approach can obtain satisfaction values around 3 times higher than the naïve strategies for the top-list data.

1. Introduction

Getting nearer to the world of *Internet of Things*, a variety of promising applications, for instance, health monitoring, product monitoring, structure monitoring, smart appliance control, and smart buildings, have emerged [1–6]. Those applications can be driven by up to petabyte scale of data streams. Not only a high volume of collected values and metadata, but variety and velocity of the streams are also challenging [7]. Such streams are continuously transmitted to the cloud service to obtain some knowledge according to user requirements. However, it is often likely that the capacity of a communication link between a gateway at monitoring field and a faraway server is limited, as illustrated in Figure 1. For example, from our previous work [8], the real-world system where the multiapplication server remotely controls the branch building in a rural area (near Toyota city) has been developed and deployed. In the building, even though a 3G network is accessible, it is not stable and causes some losses

of data due to the remote distance between base stations and the mountainous areas.

Throughout the last decade, many researchers have dedicated their contributions to handling the bottleneck issue. In service level, a data reduction could be accomplished by aggregation and compression techniques, regardless of application information, for example, in [9]. Along with reducing amount, the quality of service (QoS) such as throughputs and delays could also be specified and controlled [10]. Above the service level, the solutions on application level exploit user-specified requirements to draw only requested data streams (i.e., drop unrequested streams) [11–13]. For example, in a lighting control system of smart buildings, the data from movement and occupancy sensors are significant while data from temperature and heat detectors are ignorable. Unlike QoS in the service level, there are just a few studies on the quality of information at the application level. The key idea is that the “importance” of any specific sensing data for one application can be different for the other applications. For

FIGURE 1: Bottleneck in rural area smart building.

instance, the sensing data from smoke and gas detectors are considered as very important to the fire detecting application but nothing to the lighting control application. Also, there are many other attributes of *information quality* (IQ) that could be considered, for example, source reliability, spatiotemporal relevance, and accuracy [14]. Bisdikian et al. have described the concept of value assessment for wireless sensor networks in terms of *value of information* (VoI) [15]. VoI is an “assessment” of the utility of information (a kind of utility function) regarding innate properties (information quality, IQ). According to the assessed value, we can numerically show the importance of the data from movement and occupancy sensors for different applications such as a lighting control system and a fire detecting system. Reference [15] has also presented a framework to determine VoI based on a multiattribute decision-making technique. However, there are some challenges which have no mention yet. The first challenge of multiple requirements is that not only the different importance of properties within one application but system managers should be able to specify different priorities for different applications as well. For instance, generally in smart building systems, a controlling switching device application is supposed to have a lower priority than a security application. Therefore, the utility function should explicitly assess the sensing data by considering application priorities together with conflicting properties in each application in a best-mixed way. A framework is necessary to enable such functions. The second challenge is the delays due to the evaluation process. Such delays can cause an unsatisfied condition in real-time systems and data losses due to memory limitation at the local gateway. Thus, a lightweight decision-making is preferable.

In this study, a VoI prioritization system on requirement-based data streaming in shared wireless sensor networks (WSNs), operating on both server-side and local-gateway-side, is proposed. We focus on the following scenario. Users submit request with their priorities to the server in pairwise comparison scale. One request contains a table of conflicting

properties, considered as criteria, and multiple tables of alternatives. The criteria table presents different importance of each property for one application. The alternatives table refers to importance of the classified groups of values for each property across one another in pairwise comparison scale.

Our system mainly exploits Multicriteria Decision Analysis (MCDA) techniques to assess the importance score on each sensor data with consideration of overall user requirements. The priority queue of the gateway orders the data correspondingly to the assessed values. If the buffer queue exceeds its limit, it will drop the lowest priority item. In particular, the first-tier server will execute most of the heavy computation when users submit their requirements. The second-tier local gateways will only compute the simple classification and multiplication operations.

To evaluate our proposed system, we have introduced an example monitoring system as a practical case study on top of our previously proposed data collection system in [8]. The simulation results show that our system can achieve the best satisfaction to the given requirements, particularly, 43.29% higher over top-criterion policy and 3 times over noncriterion policy for the top-three items. We also evaluate our merging algorithm by comparing the similarity of the merged requirement and original ones from users. The results show 0.995 cosine-similarity.

The rest of the paper is organized as follows. Section 2 mentions related works, including the existing compression solutions and data prioritization. In the same section, we also settle our contributions at the end. In the same section, we settle our contributions at the end. Next, we introduce our proposed system in Section 3 along with an illustrative example. Based on the example, we set up the environment for simulation and show the results of the performance evaluation in Section 4. Section 5 introduces some discussion issues about the limitations of our works and the research directions.

2. Background and Related Works

To handle the bottleneck problem that occurred in the cloud-based IoT services, many efforts have been dedicated so far.

An optimization can be accomplished on service level or application level. The optimization on service level has no consideration about semantic meaning. It is basically about compression techniques and quality of service (QoS). On the other hand, the optimization on application level pays regard to user needs along with the data contents and contexts. There are several techniques which have been introduced, for instance, stream query, pub/sub mechanism, and contextual analysis. A few researches mention about information quality (IQ) in a similar way to QoS in the service layer. The optimization on both layers can orthogonally work. In this paper, we focus on satisfying the user-specified information quality on the application level optimization. This section gives a brief background and discussion on the existing optimization approaches.

2.1. Service Level Data Reduction and Quality Control. To compress the data, many researchers focus on encoding techniques, for example, in [8, 9]. The efficiency matrices are data size, time, and content loss. One of the known implementations is Packedobjects (PO). It is a compression library for XML format that is easy-to-implement and highly reliable and has acceptable efficiency. According to [9], using PO library can reduce the data size down to only 8% of the original size. The size of a PO-compressed-XML sensor value is about 31 bytes. Moreover, the effect of latency and packet loss is significantly low. Also, the decompression time is quite short.

Orthogonally, the term *quality of service (QoS)* represents the performance of user satisfaction in communication service. The commonly found criteria in QoS are throughput, delay, and availability. Currently, most standard communication channels have already embedded QoS-controlling supports. One example is IEEE 802.11 [10].

2.2. Application Level Data Reduction and Quality Control. The application level data reduction significantly exists in real-time systems. With user-specified requests, the data which are not included could be filtered out at an early state. This could be accomplished by many approaches, for instance, rule-based engine in [11], query-based acquisition system in [16], and publish-subscribe procedure in [12, 13].

Similarly to QoS, *information quality (IQ)* has been introduced to represent a quality of the information content [14]. For undetermined environments, like mobile ad hoc networks, most researches anticipate the IQ by some common knowledge. For instance, a scheme called *Latency and Coverage Optimized Data Collection* is introduced to collect sensing data through vehicular ad hoc networks. If it is necessary to discard some data on the way to the data center, it will consider the latency and coverage for making a decision. Each mobile vehicle will process dual-optimization under constrained situation to prioritize the arrival data and discard the less one. In the collection system proposed in [17], the potentiality of the source infers the data preference. *RushNet* in [18] determines intermittent data as urgent information while treating continuous data as delay-insensitive information. However, some researches let the source itself

notify how urgent the data it senses by some mechanism. For example, in [19], an *Adaptive Communication Control Based on a Differentiated Delay (ACCDS)* scheme is proposed to adaptively control QoS inside the sensor and actuator networks using a differentiated delay framework. The ACCDS scheme can significantly achieve network delay reduction for a delay-sensitive data and communication cost reduction for delay-insensitive interesting data.

However, not only thematic, spatiality, and source reliability, but also other properties, such as accuracy and timeliness, should be taken into consideration. A general framework has been introduced in [15] by applying a Multi-criteria Decision Analysis (MCDA) technique called Analytic Hierarchy Process (AHP). The MCDA is an approach for explicitly evaluating multiple conflicting criteria to sort for recommendation list or to make a final decision [20]. For example, you can use MCDA to help you choose between two items in the supermarket by considering both price and quality (conflicting criteria). There are many formal MCDA techniques, which are broadly accepted. The significant difference between them is how they combine the values from all criteria. One of the simplest MCDA techniques is a linear additive model. The linear additive approach arithmetically adds the multiplication results of the score and the corresponding weight for each criterion to represent the final score of each alternative. Analytic Hierarchy Process is another technique that develops the linear additive model by obtaining the weight of both criteria and options from pairwise comparison instead of independent absolute value. The AHP is another technique that develops the linear additive model by obtaining the weight of both criteria and options from pairwise comparison instead of independent absolute value. Unfortunately, AHP is confronted with a heavy criticism of undesired phenomenon called rank reversal [21]. Later, the modified version of AHP called Rembrandt prevents such a phenomenon [22–24].

According to the definition in [15], value of information (VoI) is an assessment of the utility of an information product when used in a specific usage context. The authors of [15] have also defined the taxonomy of VoI attribute in sensor networks and present how VoI depends on the quality characteristics of information (QoI) with mentioning easily derivable relations from QoI to VoI. Some researchers focus on evaluation functions for specific attributes [25, 26]. However, we presume such functions already exist. Though most of those approaches have focused on the prioritizing sensor data with multiple criteria, to the best of our survey, there is no proposal to handle several multicriteria requirements and, still, no lightweight prioritization system for real-time operating on the local gateway.

2.3. Our Contributions. There are two significant contributions of our proposal to prioritize the value of information in real-time. Firstly, our proposed system generates one merged requirement to represent several multicriteria requirements, which have different priorities, from users. The results show that the merged requirement has 99.5% similarity to the original users' requirements for prioritization. Moreover, the

FIGURE 2: System overview.

simulation results show the satisfaction efficacy of the proposed method compared to the naïve approaches. Secondly, our two-tier prioritization system can reduce computation load for prioritizing data in particular detail at the local gateway from exponential to polynomial complexity.

3. Proposed Method

This chapter gives a brief overview of the proposed system and related technical tools.

3.1. Preliminary. Throughout this paper, we use the terms “requirement” and “request” almost interchangeably to refer to the specification that specifies the satisfying information value for an application. The requirement is a ready-to-use specification, which is derived from the user-defined form of request. One request is composed of one criteria table and multiple alternatives tables. “Criteria” mean standards used to judge something. “Alternatives” are the things to be judged. In our work, there are two possible standards, which are defined as criteria: (1) application and (2) information quality (IQ) property. System managers control the first standard, that is, application, while system users can customize the second one, IQ property. We use both standards to judge the values of sensing data. Note that the term “information quality” may be replaced with the word “property” and “attribute” in some contexts.

3.2. System Overview. The proposed system relies on the scenario as depicted in Figure 2. Users submit their requests of sensing streams to the resourceful cloud server. Then, the cloud server forwards the requests to the resourceless

local gateways. The local gateways transmit the sensing streams back to the server according to the requests. The request includes multicriteria specification indicating the value of information and each one owns a different priority. In our study, *resourceless* refers to having the insufficient communication link bandwidth and gateway buffer to afford to retain and transmit all of the arrived data streams within satisfied delay in the real-time system. We note that our system is designed to work independently with the data collection process from sensing sources and the query process at the server.

Our approach achieves VoI prioritization according to various multicriteria requests with two-tier operating fashion, one on the cloud server and the other on the local gateway. In particular, the cloud server takes responsibilities on (i) merging all requests into one consistent requirement and (ii) calculating values of each merged information category on each criterion. In the same time, the local gateway is accountable for (i) categorizing information on flowing stream to one of the merged groups for each criterion, (ii) summarizing final value by multiplying all categorizing values, and (iii) prioritizing sending queue according to summarized values. With the priority queue, the highest-value data are supposed to be sent to the server first. On the other hand, it will discard the lowest-value data if the stored data size is going to exceed the buffer capacity. Most of our procedure has been achieved by tailoring Multicriteria Decision Analysis (MCDA) techniques called *Rembrandt* to multicriteria with varied-alternatives analysis instead of constant alternatives. The proposed system drives two processes. One is *request-merging* done on the cloud server. The other is *value-based selection* performed on both tiers.

FIGURE 3: Summary of Rembrandt's process.

3.3. Multicriteria Decision Analysis: Rembrandt. According to [20], *Multicriteria Decision Analysis* (MCDA in short) is an analysis for making a decision among existing alternatives against the considered criteria. People use MCDA techniques for many purposes such as identifying the most preferred option, ranking, bounding list, and distinguishing acceptable possibilities. Among a number of standard techniques, an Analytical Hierarchy Process (AHP) technique is attractive because the pairwise comparison scale is easy-to-specify. However, there are some criticisms about the AHP devised by Saaty [21]. The most significant phenomenon is called *rank reversal* [21, 22]. There are many methods that modify the original one to avoid or get rid of this phenomenon. According to [22], the most effective one is the *Rembrandt* system.

The Rembrandt system is a multiplicative version of AHP approach. It uses geometric mean for value calculation instead of the arithmetic mean to prevent rank reversal. Additionally, in [21], other benefits of the Rembrandt system have been proven. The procedure is summarized as depicted in Figure 3. To calculate final score of the alternatives (sensing data), we need the weight of each alternative J when considering individual criterion I (denoted by w_{ji}) and the weight of each criterion I , that is, importance of the criterion over all criteria (denoted by $O(i)$). Note that the weight of criterion expresses how important it is when considering all criteria.

There are four parallel steps to compute the alternatives weight, w_{ij} , and the criteria weight, $O(i)$. The first step is to change members in comparison matrices to Rembrandt (δ_{jk}) scale expressed as an exponential function of the difference between the echelons of value on the geometric scale defined by Lootsma in [23]. The second step is to transform Rembrandt-scale members into logarithm scale

as an exponential function of itself multiplied by scale parameter using $\ln \sqrt{2}$ for criteria, to obtain $O(i)$, and $\ln 2$ for alternatives, to obtain w_{ij} . The third step is to calculate a row-wise geometric mean of the transformed values. The fourth step is to additively normalize all row-wise means. The normalized values in the fourth step are $O(i)$ and w_{ij} for criteria and alternative, respectively. The final score of each alternative J across all criteria denoted by w_j is calculated by product function as formulated in

$$w_j = \prod_{i=1}^k w_{ji}^{O(i)}. \quad (1)$$

The existing framework considers one sensing data as one alternative. In other words, we cannot start assessing the score of alternatives, that is, sensing data, before they are available at local gateway. But the computation complexity for the whole assessment is in exponential scale, which is unacceptable in real-time systems. Thus, we introduce a varied-alternative MCDA framework to reduce the complexity at local gateway to linear scale. In particular, we use a set or range of possible values for each attribute as an alternative instead of individual sensing data. With such a framework, we can preliminarily compute the partial scores of any possible values for all attributes over consideration of all criteria, $w_{ji}^{O(i)}$.

3.4. Request-Merging Process. The request-merging process is a supplementary process to support several application requests from users. It merges all requests into one representative for computing a score table to submit to the gateway as a merged requirement. The merging process is performed as depicted in Figure 4(a). The inputs are (i) application comparison matrix, (ii) criteria-comparison matrix for each

FIGURE 4: Proposed system: (a) request-merging process and (b) value-based selection process. * refers to periodically recalculating the time-dependent criteria for all items in the queue.

application, and (iii) alternatives-comparison matrices in each criterion for each application.

The first step is to regroup the criteria and alternatives of all requests, that is, inputs (ii) and (iii), respectively. The regrouped set is composed of disjoint and intersection sets

from all existing sets. Note that we use the term *set* for both discrete set and continuous range. Then, we calculate the final score of each regrouped set by the modified Rembrandt algorithm when considering application comparison matrix as criteria.

The second step is to reverse the final score f to a value in the Rembrandt level. To do so, we have to use the transformation function as shown in

$$r(f) = \frac{\ln(f)}{\ln(2)}. \quad (2)$$

The third step is to create the pairwise comparison table of the regrouped set. A member m_{ab} in the comparison table is the difference between the Rembrandt level value of alternative a , denoted by $r[f(a)]$, and that of alternative b , denoted by $r[f(b)]$.

The last step is to map constructed alternatives tables to the corresponding criteria table. The *merged request* includes the constructed criteria table and mapped alternatives tables.

3.5. Value-Based Selection Process. A value-based selection process is the prioritization process according to user requests at the local gateway, shown in Figure 4(b). There are four events which can drive an action of the process. The first event is that a cloud server gets a new request from an application. Such an application must define *criteria* and corresponding *alternatives* and create a *request* including *comparison table* of the criteria as well as that of the alternatives for each criterion. Then, the cloud server will create a *requirement* from all application requests and *application comparison table*, which contains compared priority value between each pair of applications.

As explained in the request-merging process, a merged request is created to represent all user requests. Then, the merged request is used to create a *requirement*. The requirement refers to definitions for classifying alternatives in each criterion and a *reference score table*. The reference score table contains a precalculated impact score. The precalculated impact score is a row-wise geometric mean powered by row-wise geometric means of its parent criterion, pending at the weight calculation step of the Rembrandt system.

The second activating event is that sensors send their collected data to the gateway. After that, the gateway classifies the data into one of the defined alternatives for each criterion and continues the last step of the calculation. Then the gateway uses the results for prioritizing data in its sending queue by using priority queueing feature of the congestion management. Exceptionally for the time-dependent criterion, the score of sensing data must be recalculated periodically as well as the prioritization queue.

According to priorities in the sending queue, it will send the higher-priority data first when a link is available for data transmission (i.e., third event). On the other hand, it will remove the lowest priority data when its buffer is full (i.e., fourth event).

3.6. Illustrative Example. In this paper, a smart building system is used as a case study to illustrate the concept of user requests. The smart building usually concerns about enhancing energy efficiency, user comfort, and safety of residents and properties. In such a system, there are several applications sharing data streams from the sensors installed in the building. We pick up four applications as follows:

TABLE 1: Saaty's scale (refer to Appendix 5 in [20]).

Priority index	Importance of A over B
Equal	1
Moderately higher	3
Strongly higher	5
Very strongly higher	7
Overwhelmingly higher	9

HVAC, air-quality-and-window control, switching devices, and security-and-safety. Each application defines a different kind of request and holds a different priority. Table 2 shows an application comparison matrix. The application priorities are specified according to AHP scale in Table 1. For instance, the HVAC request has equal, moderately higher, and moderately low priority compared to the air quality request, the switching request, and the security request, respectively.

We define criteria by using the VoI attribute taxonomy in [15]. Without any impact on our study, we assume that all sensor data are from the trustworthy sensors and already in the compatible format. We use four criteria over the content of data as follows: thematic relevance, spatiotemporal relevance, accuracy, and timeliness. We determine the thematic relevance criterion by using the sensor types according to cross-tabulated smart building applications and sensors in [5]. In conclusion, there are five alternative sensor types: temperature sensors and heat detector, movement and occupancy sensors, smoke and gas detectors, status sensors, and glass break sensors. Each application may require different sets of sensor types. For example, an HVAC application is for monitoring the temperature and the status of building part such as opening of windows with temperature and heat detectors and movement and occupancy sensors, respectively. In the same time, an air quality application is for processing the dangerous intensity of gas in the air, for example, CO₂, from the smoke and gas detector (i.e., MEMS). Obviously, in thematic relevance criterion, the values of the relevant alternatives (i.e., sensor types) must be higher than those of the others as much as possible.

To simplify an example of spatiotemporal criterion, we assume all sensors are statically installed. So, this criterion will leave just spatial dimension. Table 3 shows HVAC application request. The request consists of a comparison matrix of criteria, followed by those of alternatives. HVAC is an application for providing indoor thermal comfort and acceptable air quality. It usually concerns the following criteria: *thematic*, *accuracy*, and *timeliness*. According to Saaty's scale, the criteria table of the HVAC request shows the priority of the thematic one-step higher, somewhat more important, than timeliness and two-step higher, much more important, than accuracy. For each criterion, the possible values are grouped and defined as an alternative. For example, thematic consists of two alternatives, *Set 1*, which represents the considered types of sensors like *temperature and heat* and *movement and occupancy* and *others* (i.e., the other types of sensors). The possible values of accuracy are in the range of floating number from 0 to 1. According to the accuracy table,

TABLE 2: Applications comparison table.

Application	HVAC	Air quality	Switching	Security
HVAC	1	1	3	1/3
Air quality	1	1	3	1/3
Switching	1/3	1/3	1	1/5
Security	3	3	5	1

TABLE 3: HVAC request.

(a)				
Criteria	Thematic	Accuracy	Timeliness	Others
Thematic	1	5	3	9
Accuracy	1/5	1	5	9
Timeliness	1/3	3	1	5
Others	1/9	1/3	1/5	1
(b)				
Thematic			Set 1	Others
Set 1 = {Temperature&heat, Movement&occupancy}			1	9
Others			1/9	1
(c)				
Accuracy	High	Acceptable		Low
High [0.8, 1]	1	3		5
Acceptable [0.6, 0.8)	1/3	1		3
Low [0, 0.6)	1/5	1/3		1
(d)				
Timeliness	High	Medium		Low
High ratio [0.7, —)	1	3		7
Medium ratio [0.3, 0.7)	1/3	1		5
Low ratio [0, 0.3)	1/7	1/5		1

the possible values are grouped into three alternatives: *high*, *acceptable*, and *low*. We define the timeliness value as a ratio of the amount of data that is selected and sent to the amount of generating data at a specific time window. Thus, it is not static but dynamically changes over the time.

From the definition mentioned above, we create the input requests, including application comparison table, HVAC request, air-quality-and-window-control request, switching device request, and security-and-safety request. The last three application requests are shown in Tables 4, 5, and 6, respectively.

4. Experimental Results

We have used a network simulator Scenargie version 1.8 for simulating our scenario. Its powerful GUI and precise modeling of protocol sets are quite beneficial to a variety of simulator users. We have implemented the gateway-side computation (the value aggregation, sorting, and selection algorithms) in C++ and incorporated them into the simulator code. However, the server-side computation (the merging

algorithm) is implemented in Java considering future usage in real systems.

4.1. Simulation Setup. To evaluate our system, we have applied our method to the smart building system described in Section 3.6 and set up the significant inputs of the simulation including sensor data, application requirements, and resource constraints as follows.

4.1.1. Sensor Data. For the sensor data, the descriptions that are related to value assessment consist of sensor IDs, types, sampling rates, and accuracy. We generate data from five types of 200 sensors with the same sampling rate (100 ms^{-1}). Each type has the same number of sensors (i.e., 40 sensors for each type). The sensor IDs are running numbers from 1 to 200 by the following type orders: (i) temperature and heat, (ii) movement and occupancy, (iii) smoke and gas, (iv) status, and (v) glass break. Then, we shuffle the data sequence and randomly assign accuracy of the sensors.

4.1.2. Application Requirements. A merged requirement consists of definitions for classifying regrouped alternatives in

TABLE 4: Air-quality-and-window-control request.

(a)					
Criteria	Thematic	Spatiality	Accuracy	Timeliness	Others
Thematic	1	1	5	3	9
Spatiality	1	1	5	3	9
Accuracy	1/5	1/5	1	1/3	3
Timeliness	1/3	1/3	3	1	5
Others	1/9	1/9	1/3	1/5	1
(b)					
Thematic	Set 1				Others
Set 1 = {Movement&occupancy, Smoke&gas, Status}	1				9
Others	1/9				1
(c)					
Spatiality	Set 1		Set 2		Others
Set 1 = {41–50}	1		3		5
Set 2 = {81–100, 121–130}	1/3		1		3
Others	1/5		1/3		1
(d)					
Accuracy	High		Acceptable		Low
High [0.8, 1]	1		3		5
Acceptable [0.6, 0.8)	1/3		1		3
Low [0, 0.6)	1/5		1/3		1
(e)					
Timeliness	High		Medium		Low
High ratio [0.7, —)	1		3		5
Medium ratio [0.3, 0.7)	1/3		1		3
Low ratio [0, 0.3)	1/5		1/3		1

each mapped criterion and a reference score table of the merged request. An individual application request is defined in Section 3.6. After applying the merging algorithm as described in Section 3, we obtain a merged requirement and a reference score table as shown in Tables 7 and 8, respectively. We use the merged requirement as an input at the gateway for making a selection decision in the simulation.

However, timeliness is an exceptional criterion. According to the previous section, we define the timeliness as the selection frequency of streams. To enhance the overall score, we reverse the score of timeliness at the gateway for encouraging the rarely selected stream to be selected. The new score must range between the minimum score and the maximum score of the timeliness criterion to keep the same impact to final score as before reversing. Thus, we formulate a reverse function, f , of x regarding the highest impact alternative (HIA) and the lowest impact alternative (LIA) as shown in (3). Note that $\text{score}(x)$ denotes the score of alternative x in the reference table.

$$f(x) = \text{score}(\text{LIA}) + \frac{\text{score}(\text{HIA}) - \text{score}(x)}{\text{score}(x)}. \quad (3)$$

4.2. Simulation Results. In our experiments, we define the term *satisfaction value* as an evaluation metric. The satisfaction value of each sensing data item is calculated as shown in (4). Note that match_score_i is a precalculated score of matching alternative belonging to criterion i of that sensing data according to the reference score table and max_score is the production result of maximum scores from all criteria.

$$\text{satisfaction value} = \frac{\prod_{i=1}^k \text{match_score}_i}{\text{max_score}} \times 100\%. \quad (4)$$

The satisfaction value will be 100% when that sensor data entry matches to the highest-score alternative for all criteria. Sending all data does not mean the satisfaction value will become one which reflects the fact that not all generated data are needed. We compared our method to the traditional First-In-First-Out method (*noncriteria*: NC) and the method that uses only the highest impact criteria (*top-criterion*: TC).

To show the performance of our merging process, we compare the similarity of the prioritized sequence of all possible sensing data in the case study from the merged requirement with that from the original requirements. Note that a set of possible sensing data items is equivalent to a set of possible combinations of alternatives from all criteria. We

TABLE 5: Switching-device request.

(a)				
Criteria	Thematic	Spatiality	Accuracy	Others
Thematic	1	1	5	9
Spatiality	1	1	5	9
Accuracy	1/5	1/5	1	3
Others	1/9	1/9	1/3	1
(b)				
Thematic	Set 1			Others
Set 1 = {Movement&occupancy, Status}	1			9
Others	1/9			1
(c)				
Spatiality	Set 1	Set 2		Others
Set 1 = {45–60}	1	3		5
Set 2 = {131–140}	1/3	1		3
Others	1/5	1/3		1
(d)				
Accuracy	High	Acceptable		Low
High [0.8, 1]	1	3		5
Acceptable [0.6, 0.8)	1/3	1		3
Low [0, 0.6)	1/5	1/3		1

TABLE 6: Security-and-safety request.

(a)				
Criteria	Thematic	Spatiality	Accuracy	Others
Thematic	1	1	3	9
Spatiality	1	1	3	9
Accuracy	1/3	1/3	1	7
Others	1/9	1/9	1/7	1
(b)				
Thematic	Set 1			Others
Set 1 = {Temperature&heat, Movement&occupancy, Smoke&gas, Status, Glass break}			1	9
Others			1/9	1
(c)				
Spatiality	Set 1			Others
Set 1 = {1–20, 151–160, 181–200}			1	9
Others			1/9	1
(d)				
Accuracy	Extreme	High	Acceptable	Low
Extremely high [1, 0.9]	1	3	5	9
High [0.8, 0.9)	1/3	1	3	7
Acceptable [0.6, 0.8)	1/5	1/3	1	5
Low [0, 0.6)	1/9	1/7	1/5	1

TABLE 7: Merged requirement.

(a)							
Criteria	Accuracy	Spatiotemporal		Thematic		Timeliness	
Accuracy	1	0.36		0.23		3.15	
Spatiotemporal	2.81	1		0.39		4.97	
Thematic	4.37	2.56		1		6.53	
Timeliness	0.32	0.20		0.15		1	
(b)							
Thematic							
Glass break	1	0.18	0.36	0.27	0.36	4.58	
Movement and occupancy	5.46	1	3.68	2.79	3.68	9.05	
Smoke and gas	2.79	0.27	1	0.53	1	6.37	
Status	3.68	0.36	1.89	1	1.89	7.26	
Temperature and heat	2.79	0.27	1	0.53	1	6.37	
Others	0.22	0.11	0.16	0.14	0.16	1	
(c)							
Accuracy							
R: [0.0, 0.6)	1	0.26		0.17		0.15	
R: [0.6, 0.8)	3.91	1		0.33		0.26	
R: [0.8, 0.9)	5.92	3.01		1		0.53	
R: [0.9, 1.0]	6.81	3.90		1.89		1	
(d)							
Timeliness							
R: [0.0, 0.3)	1			0.25		0.17	
R: [0.3, 0.7)	4.02			1		0.33	
R: [0.7, —]	6.03			3.01		1	
(e)							
Spatiotemporal							
{131–140}	1	0.33	0.54	0.41	0.78	0.78	1.28
{1–20, 151–160, 181–200}	3.00	1	2.14	1.57	2.72	2.72	3.29
{41–44}	1.86	0.47	1	0.64	1.57	1.57	2.14
{45–50}	2.43	0.64	1.57	1	2.14	2.14	2.72
{51–60}	1.29	0.37	0.64	0.47	1	1	1.57
{81–100, 121–130}	1.29	0.37	0.64	0.47	1	1	1.57
Others	0.78	0.30	0.47	0.37	0.64	0.64	1

straightforwardly calculate the final score of each alternative according to the original requirements by the following steps. The first step is to find the final scores of all alternatives when considering only individual requirements using modified Rembrandt system explained in Section 3.3. The second step is to combine the final score of all alternatives from all requirements considering the priority of each requirement by (1). We plot the prioritized sequences from the merged requirement on the horizontal axis and the other one from the multiple user requirements on the vertical axis, as shown in Figure 5. The results show that the cosine-similarity between both sequences is up to 0.995 (84.2°) and Pearson's correlation is up to 0.979. We also observe that there is smaller difference around high-rank and low-rank data compared to the middle-rank ones.

If we allow users to specify more criteria, the satisfaction values are significantly improved especially in the resource-insufficient environment. To present this fact, we conduct two simulation scenarios. One has sufficient buffer size but limits the available bandwidth. The other has sufficient bandwidth but limits the buffer size.

For the first scenario, we limit the available bandwidth to 128 kbps, 256 kbps, 384 kbps, and 512 kbps. The result of average satisfaction value from the total runtime is presented in Figure 6. As expected, the average satisfaction values of both *multicriteria* (MC) and *top-criterion* (TC) get lower when the resources grow larger because more data are transmitted. In other words, the gateway can transmit more low-value data. At 512 kbps, it can transmit all arrived data. We also observe the considerable difference between using

TABLE 8: Reference score table of merged requirement.

Criteria	Alternatives	Score
Thematic	Movement and occupancy	3.043
	Status	1.658
	Smoke and gas	1.225
	Temperature and heat	1.225
	Glass break	0.668
	Others	0.198
Spatiality	{1–20, 151–160, 181–200}	1.306
	{45–50}	1.166
	{41–44}	1.041
	{81–100, 121–130}	0.930
	{51–60}	0.930
	{131–140}	0.879
	Others	0.830
Accuracy	R: [0.9, 1]	1.288
	R: [0.8, 0.9)	1.172
	R: [0.6, 0.8)	0.949
	R: [0, 0.6)	0.698
Timeliness	R: [0.7, —)	1.125
	R: [0.3, 0.7)	1.017
	R: [0, 0.3)	0.874

FIGURE 5: Similarity and correlation of merging algorithm.

FIGURE 6: Satisfaction value over variation of bandwidth.

multiple criteria, using only a top-criterion, and not using any criteria in the resource-insufficient scenario.

At 128 kbps, the average satisfaction value of proposed MC approach achieves 73% and 19% increment from TC and FIFO, respectively. In particular, Figure 7 shows the amount of sensor data in each range of satisfaction value with an accumulated line. An accumulated line of FIFO grows slowly at high satisfaction and fast at low. On the contrary, the accumulated line of MC method has a high growth at high values until moderate and then keeps flat. The growth trend of the TC method is similar to that of the MC. However, the slope in the middle is shallower.

Furthermore, the average value of the top-three data items, listed in Table 9, from MC method is 43.29% and about 200% higher than MC and FIFC methods, respectively. Note that the Data ID (DID) refers to the order of the data assigned by the counter at the gateway and SID is an abbreviation of Sensor ID. We also notice that the top-three data items of both MC method and TC method contain only one sensor type, movement and occupancy, while FIFO method transmits many types of sensor data. We found that the score of movement and occupancy is about 2 times higher than the others in the example requirements. However, unlike TC, the proposed MC method still gives a chance for the data with other types to be selected if they have the higher score from other criteria.

For the second scenario, we restrain the buffer size to support 12.5%, 25%, 37.5%, 50%, 62.5%, 75%, and 87.5% of the total number of sensor streams. As shown in Figure 8, the averages of discarded satisfaction values using multicriteria are always less than that of others. Similarly to the first scenario, the results of using top-criterion method reside

FIGURE 7: Accumulated graph over satisfaction values when bandwidth is low. The values on x -axis starting from 100 down to 0 are intended to show the accumulated values from good to bad.

TABLE 9: First three data items sent to the server when limiting bandwidth at 128 kbps for each method.

(a)					
Multicriteria method					
DID	SID	Sensor type	Accuracy	Time	Value
44	45	Movement & occupancy	0.8	0.1	63.16
107	50	Movement & occupancy	0.8	0.1	63.16
79	43	Movement & occupancy	0.8	0.1	56.42
Average value = 60.91					
(b)					
Top-criterion method					
DID	SID	Sensor type	Accuracy	Time	Value
199	76	Movement & occupancy	0.7	0.1	36.41
159	57	Movement & occupancy	0.8	0.1	50.36
158	54	Movement & occupancy	0.6	0.1	40.76
Average value = 42.51					
(c)					
Noncriteria method					
DID	SID	Sensor type	Accuracy	Time	Value
0	175	Glass break	0.7	0.1	7.99
1	148	Status	0.8	0.1	24.52
2	141	Status	0.9	0.1	26.94
Average value = 19.82

FIGURE 8: Discarded satisfaction values over variation of buffer size.

FIGURE 9: Comparison of gained and dropped value when buffer is small.

between those of multicriteria and noncriteria methods. We also observe that the larger the buffer the gateway has, the lower the average values it will lose.

In the low-buffer experiment, the bar graph in Figure 9 shows the comparison of gained and dropped values from the three approaches. The worst method, noncriteria, causes the gained value to be nearly the same as the dropped value. For the multicriteria and top-criterion methods, both of them have high gained values and low dropped values. Nevertheless, we do not recommend the top-criterion method even though it seems to result in high efficiency. Because when the top-criterion is not time-dependent, some data may fall into the starvation state. With the simulation given in this paper where the top-criterion is thematic, the selected data on the low bandwidth condition mostly contain one type, movement and occupancy, which is the highest-impact

alternative. Moreover, according to the numerical value, we have also found that using multicriteria method always gains better value and loses lower value compared to the results from the top-criterion method.

5. Discussion

To control information qualities, an assessment may be executed individually for each sensing data item is entirely for each stream source. The latter can reduce the complexities of data selection and bandwidth management. However, it may lose precision to assess some attributes which are not relevant to the source. So, the tradeoff between the precision of satisfaction and complexity should be considered. Also, it is still open to study more about utilizing the well-known techniques for recommendation systems on data streaming from wireless sensor networks to satisfy user demands. To reduce complexity and keep acceptable precision, we group the possible sensing value for each attribute and precompute the reference score table.

The complexity to evaluate each sensing data depends on the number of considered attributes and the number of groups for each attribute. Thus, in the real-time system, we can guarantee the satisfied delays at the gateway by limiting the number of the groups. Furthermore, our designed system can operate in dynamic networks. Not only user requirements but also sensing sources can change during the operating time because the value assessment does not refer to the source identifiers.

The proposed method can also work together with other optimization techniques, for instance, data compression, stream query, data-centric pub/sub mechanism, or data collection system in ad hoc networks at monitoring area. Data compression approaches can be executed before sending selected data. In the same time, the rest techniques can be performed before applying our selection. The proposed method aims at improving user satisfaction of information quality in the scenario that the resources are still insufficient even if the other orthogonal optimizing techniques are already in use.

However, there are a few issues that we would like to mention for the future improvements due to user-defined requirements. The first issue is exaggerated satisfaction values due to low-varied requirements. Practically, the actual satisfaction loss cannot be directly measured. However, we suggest to apply *queue theory* technique [27] to estimate potential satisfaction loss for each requirement as feedback to users. If they show too much satisfaction loss, users are supposed to adjust their requirements to differentiate grouping more explicitly.

The second issue is user permissions in highly secured systems. In our procedure, users have to define the comparison values between each group and each attribute. The current system is not valid if users are not allowed to submit their requirements directly. We are now considering an additional feature that can generate a requirement, compatible with our system, automatically. However, it still allows users to send

some feedback, in the direct or indirect way, for adaptation during the operating period.

The third issue is the evaluation metric, satisfaction value. In this paper, we assume that the most satisfying sensing data will have 100% satisfaction value. Our defined satisfaction function considers only the score specified by users and the priorities of each application. However, the most satisfying data may not be limited to those values. Considering only specified priority can cause starvation of low priority application. To avoid that, the preferred function to compute the satisfaction value should take the balance of all applications into consideration as well.

Also, from the last scatter diagram, our merging algorithm still has a gap between the plot points and a trend line in the middle sequence. In the future work, we plan to find out more appropriate transformation function which is still less complex but more precise.

6. Conclusion

Our study has focused on maximizing satisfaction on information quality according to several, priority-various multicriteria requirements using the lightweight prioritizing mechanism on local gateway before forwarding to the online server via resource-limited links. We have proposed a two-tier prioritization system tailoring Multicriteria Decision Analysis technique called Rembrandt. With our proposed approach, all heavy computations, including transforming logarithm scale and finding row-wise geometric means for requirement-merge and for categorizing score, are preliminarily executed at the online server and only polynomial-complex operations, including conditional matching and final score multiplication, are left at the local gateway. We present our system with an illustrative case study of the smart building. Our merging mechanism obtains one representative requirement that achieves 0.995 cosine-similarity of satisfaction value compared to that of all originally inputted requirements. Simulation results affirm that applying multicriteria requirements can provide much higher satisfaction value compared to the naïve methods. In particular, the top-three sensing data items from our multicriteria approach have about 1.4 times higher average satisfaction values for those from the one-top-criterion approach and 3 times for those from the noncriterion approach.

Disclosure

Part of this work was carried out under the Cooperative Research Project Program of RIEC, Tohoku University.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by JSPS KAKENHI (JP15H02690, JP26220001, and JP26220001).

References

- [1] D. Puccinelli and M. Haenggi, "Wireless sensor networks: applications and challenges of ubiquitous sensing," *IEEE Circuits and Systems Magazine*, vol. 5, no. 3, pp. 19–31, 2005.
- [2] R. K. Pragnya and J. K. Chaitanya, "Wireless sensor network based healthcare monitoring system for homely elders," *International Journal of Advances in Engineering & Technology*, vol. 6, no. 5, pp. 2078–2083, 2013.
- [3] N. K. Suryadevara and S. C. Mukhopadhyay, "Wireless sensor network based home monitoring system for wellness determination of elderly," *IEEE Sensors Journal*, vol. 12, no. 6, pp. 1965–1972, 2012.
- [4] G. Virone, A. Wood, L. Selavo et al., "An assisted living oriented information system based on a residential wireless sensor network," in *Proceedings of the 1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare (D2H2 '06)*, pp. 2–4, Arlington, Tex, USA, April 2006.
- [5] OECD, "Smart sensor networks: technologies and applications for green growth," OECD Digital Economy Papers 167, OECD Publishing, Paris, France, 2009.
- [6] Y. Zhang, R. Yu, S. Xie, W. Yao, Y. Xiao, and M. Guizani, "Home M2M networks: architectures, standards, and QoS improvement," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 44–52, 2011.
- [7] S. Suthaharan, "Big data classification: problems and challenges in network intrusion prediction with machine learning," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 4, pp. 70–73, 2014.
- [8] K. Yoi, H. Yamaguchi, A. Hiromori et al., "Multi-dimensional sensor data aggregator for adaptive network management in M2M communications," in *Proceedings of the 14th IFIP/IEEE International Symposium on Integrated Network Management (IM '15)*, pp. 1047–1052, Ottawa, Canada, May 2015.
- [9] J. N. Bagale, A. Shiyanbola, J. P. T. Moore, and A. D. Kheirkhahzadeh, "Towards a real-time data sharing system for mobile devices," in *Proceedings of the 8th International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST '14)*, pp. 147–152, Oxford, UK, September 2014.
- [10] Q. Ni, L. Romdhani, and T. Turletti, "A survey of QoS enhancements for IEEE 802.11 wireless LAN," *Wireless Communications and Mobile Computing*, vol. 4, no. 5, pp. 547–566, 2004.
- [11] Y. Sun, T.-Y. Wu, G. Zhao, and M. Guizani, "Efficient rule engine for smart building systems," *IEEE Transactions on Computers*, vol. 64, no. 6, pp. 1658–1669, 2015.
- [12] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: scalable coordination in sensor networks," in *Proceedings of the 5th Annual International Conference on Mobile Computing and Networks (MobiCom '99)*, pp. 263–270, Seattle, DC, USA, 1999.
- [13] U. Hunkeler, H. L. Truong, and A. Stanford-Clark, "MQTT-S—a publish/subscribe protocol for wireless sensor networks," in *Proceedings of the 3rd IEEE/Create-Net International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08)*, pp. 791–798, Bangalore, India, January 2008.
- [14] F. Naumann and K.-U. Sattler, Information quality: Fundamentals, techniques, and use, Tutorial at EDBT, 2006 <https://pdfs.semanticscholar.org/2434/ed992d6e62b2c3865ab275f02dfbb0fb17c7.pdf>.

- [15] C. Bisdikian, L. M. Kaplan, and M. B. Srivastava, "On the quality and value of information in sensor networks," *ACM Transactions on Sensor Networks*, vol. 9, no. 4, article 48, 2013.
- [16] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TinyDB: an acquisitional query processing system for sensor networks," *ACM Transactions on Database Systems*, vol. 30, no. 1, pp. 122–173, 2005.
- [17] Q. Xiang, X. Chen, L. Kong, L. Rao, and X. Liu, "Data preference matters: a new perspective of safety data dissemination in vehicular ad hoc networks," in *Proceedings of the 34th IEEE Annual Conference on Computer Communications and Networks (IEEE INFOCOM '15)*, pp. 1149–1157, Kowloon, Hang Kong, May 2015.
- [18] C.-J. M. Liang, K. Chen, N. B. Priyantha, J. Liu, and F. Zhao, "RushNet: practical traffic prioritization for saturated wireless sensor networks," in *Proceedings of the 12th ACM Conference on Embedded Networked Sensor Systems (SenSys '14)*, pp. 105–118, Memphis, Tenn, USA, November 2014.
- [19] Y. Xu, X. Chen, A. Liu, and C. Hu, "A latency and coverage optimized data collection scheme for smart cities based on vehicular ad-hoc networks," *Sensors*, vol. 17, no. 4, article 888, 2017.
- [20] Great Britain and Department for Communities and Local Government, *Multi-Criteria Analysis: A Manual*, Communities and Local Government, London, UK, 2009.
- [21] H. Maleki, "Comparison of the REMBRANDT system with the Wang and Elhag approach: a practical example of the rank reversal problem," *African Journal of Business Management*, vol. 6, no. 1, 2012.
- [22] D. L. Olson, G. Fliedner, and K. Currie, "Comparison of the REMBRANDT system with analytic hierarchy process," *European Journal of Operational Research*, vol. 82, no. 3, pp. 522–539, 1995.
- [23] F. Lootsma, "The REMBRANDT system for multi-criteria decision analysis via pairwise comparisons or direct rating," Tech. Rep., Faculty of Technical Mathematics and Informatics, Delft University, Delft, Netherlands, 1992.
- [24] M. B. Barfod and S. Leleur, "Scaling transformation in the rembrandt technique: examination of the progression factors," *International Journal of Information Technology and Decision Making*, vol. 12, no. 5, pp. 887–903, 2013.
- [25] G. Tychogiorgos and C. Bisdikian, "Selecting relevant sensor providers for meeting 'your' quality information needs," in *Proceedings of the 12th IEEE International Conference on Mobile Data Management (MDM '11)*, pp. 200–205, Lulea, Sweden, June 2011.
- [26] G. Tychogiorgos and C. Bisdikian, "A framework for managing the selection of spatiotemporally relevant information providers," in *Proceedings of the 2013 IFIP/IEEE International Symposium on Integrated Network Management (IM '13)*, pp. 403–410, Ghent, Belgium, May 2013.
- [27] D. Gross, J. F. Shortle, and J. M. Thompson, *Fundamentals of Queueing Theory*, John Wiley & Sons, Hoboken, NJ, USA, 4th edition, 2008.

Hindawi

Submit your manuscripts at
<https://www.hindawi.com>