PAPER Special Section on the Architectures, Protocols, and Applications for the Future Internet

A P2P Sensor Data Stream Delivery System That Guarantees the Specified Reachability under Churn Situations

Tomoya KAWAKAMI^{†a)}, Nonmember, Tomoki YOSHIHISA^{††}, and Yuuichi TERANISHI^{††,†††}, Members

SUMMARY In this paper, we propose a method to construct a scalable sensor data stream delivery system that guarantees the specified delivery quality of service (i.e., total reachability to destinations), even when delivery server resources (nodes) are in a heterogeneous churn situation. A number of P2P-based methods have been proposed for constructing a scalable and efficient sensor data stream system that accommodates different delivery cycles by distributing communication loads of the nodes. However, no existing method can guarantee delivery quality of service when the nodes on the system have a heterogeneous churn rate. As an extension of existing methods, which assign relay nodes based on the distributed hashing of the time-to-deliver, our method specifies the number of replication nodes, based on the churn rate of each node and on the relevant delivery paths. Through simulations, we confirmed that our proposed method can guarantee the required reachability, while avoiding any increase in unnecessary resource assignment costs.

key words: sensor data, data stream, delivery cycle, distributed processing, replication, churn resilience

1. Introduction

The Internet of Things (IoT) means that objects, such as computing devices, electrical appliances, and sensors, connect to the Internet and interact with each other, to collaborate and provide various intelligent services. Sensors, in particular, have an important role in the IoT, and generate observed data periodically. The continuous periodical data generated by such sensors is called the "sensor data stream." In IoT services, a huge number of sensor data streams are typically required in order to ensure data delivery to appropriate destinations such as users and processes. In such sensor data stream delivery, destinations may require different delivery cycles for the same sensor data stream, for various reasons, such as the performance of the receiver, network environments, or applications. Here, the word "delivery cycle" means an interval of data transmission [1]. The minimum unit is different by assumptions such as a second, minute, and hour. In addition, the longer delivery cycle has less data transmissions per unit time. For example, the following delivery configurations are possible:

Manuscript received August 23, 2018.

- [†]The author is with the Graduate School of Science and Technology, Nara Institute of Science and Technology, Ikoma-shi, 630– 0192 Japan.
- ^{††}The authors are with the Cybermedia Center, Osaka University, Ibaraki-shi, 567–0047 Japan.
- ^{†††}The author is with the National Institute of Information and Communications Technology, Koganei-shi, 184–8795 Japan.
 - a) E-mail: kawakami@is.naist.jp
 - DOI: 10.1587/transinf.2018NTP0014

- The live video of a solar eclipse is delivered at 30 fps to personal computer users connected to the Internet through a wire, but at 10 fps to mobile computer users connected to the Internet through a 3G channel.
- When a variety of computers forecast the temperature from existing temperature data, the delivery cycle to the different destination computers is determined based on the processing speed.
- When a user is continuously checking the amount of rain, to decide the timing for going out on a rainy day, the data are delivered once per second to a smart phone connected to a power source, but only once per minute (to reduce power consumption) if the phone is not so connected.

It is typical, in sensor data stream delivery, that sensor data gained by one sensor is shared by a large number of users. Currently, various P2P-based techniques for dispersing the communication load of the deliverer (source) have been studied in the data streaming [2]–[12]. In these studies, when the same sensor data stream is delivered to a number of terminals (destinations), the communication load of the source is dispersed by the destinations sending the received data to other destinations. When the delivery cycles differ, the sensor data stream whose delivery cycle is a common divisor of the required cycles can be delivered to all the destinations if the delivery cycles are in a multiple relationship or can be approximated as having a multiple relationship. However, the destinations still receive redundant data which are not included in the times of each required cycle.

We have proposed P2P-based methods for constructing a scalable and efficient sensor data stream system that accommodates different delivery cycles by distributing the communication loads of the nodes [13]. In the existing methods, destinations with a long delivery cycle relay the sensor data stream to other destinations, so that the load on the source is dispersed. In addition, we have proposed a method that enhances the robustness of the delivery system by replication of processing nodes [14]. However, the existing methods do not specify the appropriate number of replication nodes, and cannot guarantee the specified delivery quality of service (QoS) (i.e., total reachability to destinations) when the delivery server resources (nodes) on the system have a heterogeneous churn rate.

This paper is an extension of our previous work [15]. Our previous work [15] constructs a scalable sensor data stream delivery system that guarantees the specified reacha-

Manuscript revised December 17, 2018.

Manuscript publicized February 6, 2019.

bility as the QoS of the delivery, even when the nodes are in a heterogeneous churn situation. Compared to our previous paper, this paper shows new experiment results and comparison with related work. The contributions of our previous work and this paper are the followings;

- We propose an extension of the existing distributed data delivery method that enables high scalability by assigning relay nodes based on the distributed hash of the time-to-deliver.
- We also propose an adaptive replication method by specifying the number of replication nodes based on the churn rate of each node and on the relevant delivery paths.

The problems addressed by this paper are described in Sect. 2. The proposed method is described in Sect. 3, and its evaluation is summarized in Sect. 4. We describe the related work in Sect. 5, and conclude the paper in Sect. 6.

2. Problems Addressed

2.1 Assumptions

In the sensor data stream delivery system that we assume in this paper, computers (nodes) relaying sensor data streams construct a P2P overlay network. The sensor data stream delivery system distributes the delivery loads to the nodes, and maintains high scalability in environments where there are a huge number of sensor data streams and destinations. Sensor data streams are periodically sent from their sources through the Internet, and delivered to destinations by hops among nodes. Destinations request sensor data streams with those delivery cycles to a specific node, also through the Internet. We assume that each delivery system determines the selectable delivery cycles in advance, and around 10 patterns of the selectable delivery cycles are enough to satisfy user demand in most of real situations. For example in live streaming, YouTube Live[†] supports 11 video qualities from 240p to 2160p with 60fps, and also OBS Studio^{††} supports 11 resolutions from 480×300 to 1440×900 in its version 22. Nodes are able to send sensor data to other nodes at any time, and sensor data are distributed for each sensor data stream and time.

The sensor data streams are S_i $(i = 1, \dots, l)$, destinations are D_i $(i = 1, \dots, m)$, and nodes are N_i $(i = 1, \dots, n)$. Figure 1 shows a model of the delivery system. Here, the number of sensor data streams is l = 2, the number of destinations is m = 4 and the number of nodes is n = 3. The 'a' represents the sensor data stream S_1 from a temperature sensor, and the 'b' represents the sensor data stream S_2 from a live camera. The delivery cycles are shown near the sources, nodes, and destinations in Fig. 1. The 's' represents the source of the sensor data stream, and the numbers near destinations are the requested delivery cycles from each destination. For example, a temperature sensor represented



Fig.1 System model

as S_1 acquires a temperature data once every second, and D_1 receives the data once every second, D_2 and D_3 receives the data once every two seconds, and D_4 receives the data once every three seconds. In addition, When a destination does not request the sensor data stream, the delivery cycle is 0. This corresponds to the case, for example, where a live camera represented as S2 acquires an image once every second, and D_1 does not view the image, D_2 and D_3 view the image once every second, and D_4 views the image once every three seconds. In this paper, we assume that each delivery system determines the selectable delivery cycles in advance, and they are represented by C_i ($i = 1, 2, \dots$). The sensor data delivery system assigns the delivery cycles or times to relay sensor data streams to nodes, and the nodes send and receive various sensor data to and from each other at specific times.

2.2 Replication of Assigned Nodes

Currently, we have proposed a P2P-based method that delivers sensor data to a huge number of destinations with heterogeneous delivery cycles [13]. The proposed method determines the relay nodes based on distributed hashing, and each node constructs delivery paths autonomously.

In the sensor data stream delivery, the number of data items to send/receive varies among different delivery cycles, and the shorter the delivery cycle, the larger the number of data items and the load. Therefore, the existing method using distributed hashing first generates circular hash spaces for each sensor data stream and puts nodes in the hash spaces based on the distributed hashing of the combination of sensor data stream and node ID. Then, the method divides each hash space into partial hash spaces as groups for each delivery cycle, to ensure that the partial hash space of the shorter cycle will have more nodes. The size of each partial hash space is determined based on its cycle. For example, when the selectable delivery cycles are $C_i = i$ (i = 1, 2, 3), the ratio of the sizes of partial hash spaces is $1/C_1 : 1/C_2 : 1/C_3 = 1/1 : 1/2 : 1/3 = 6 : 3 : 2$. The

[†]https://www.youtube.com/live (accessed Dec. 1, 2018)

^{††}https://obsproject.com/ (accessed Dec. 1, 2018)



Fig. 2 Assignment to a group of cycle

method treats each partial hash space as circular, and assigns related times for each cycle to nodes in its partial hash space. When there are no nodes in the partial hash space, the method assigns the partial hash space to the nearest neighbor node in the next partial hash space. In addition, the method determines the root node in the partial hash space of the shortest cycle based on distributed hashing, such as the least common multiple of cycles. The root node receives data first from the source of the sensor data stream.

We have proposed a method that enhances the robustness of the delivery system with a successor list used in Chord [14], [16]. Successors are nodes located next to the assigned node, and the successor list is used for node and data replication. Figure 2 shows an example where the number of nodes is n = 8, cycles are $C_i = i$ (i = 1, 2, 3), the size of a hash space is 2^p , and the length of the successor list is 2. Here, "group of cycle" in Fig. 2 means a group whose members are assigned the same delivery cycle. In addition, the "p" means a bit length of the hash space shared among all nodes. The less p causes more collisions of the assigned nodes and disturbs load distribution. Therefore, the p requires to be appropriately determined in advance based on the assumed number of nodes. The beginning values of each partial hash space in Fig. 2 are $2^p \times 0/11$, $2^p \times 6/11$, and $2^p \times 9/11$. In the existing method, the number of replication nodes is static at a specific value. However, real systems typically target a specific reliability as QoS (i.e., reachability to destinations). In this paper, we call the reliability, "targeting system reliability," and assume that this is known at each specified time. SR_t denotes the targeting system reliability at time t. The appropriate number of replication nodes for the targeting system reliability changes with the cycle groups and times. If the number of replications is static, cases that do not satisfy the targeting system reliability and/or increase unnecessary costs occur. Thus, in this paper, we aim to reduce the number of replication nodes satisfying the targeting reliability at each specified time.

3. Node Replication Method in the Sensor Data Stream Delivery System

In this paper, we propose a node replication method based on environmental variables such as targeting system reliability, the churn rates of the respective nodes, and the number of destinations for each delivery cycle.

3.1 Concept

We assume that the delivery system can obtain the elements below, and that each node determines its own replication nodes at each specified time. Each node determines its own replication nodes based on probabilistic calculations, and the expected value of the number of reachable destinations satisfies the targeting system reliability.

• Targeting system reliability

In this paper, we define the reliability as the probability that sensor data reach the requesting destination, or the ratio of reachable destinations to all requesting destinations. We assume that the provider of the delivery system determines the targeting system reliability to guarantee the specified QoS. The number of replication nodes varies with the targeting system reliability; for example, if the targeting system reliability increases, nodes send replicated data to more nodes at each specified time, to achieve this increased reliability.

• Churn rates of the respective nodes

The churn rates of the original node and replication node candidates affect to the determined replication nodes; for example, if the replication node candidates have low churn rates, the targeting system reliability can be achieved by only a few replication nodes, compared to the case of high churn rates. In this paper, we assume that each node knows its own churn rate and the approximate churn rates of the replication node candidates based on the specific model such as the bathtub curve [17]. The bathtub curve is widely used in reliability engineering, and Backblaze shows the disk failure rates shaped to the bathtub curve [18]. Therefore, it is enough probable to estimate the churn rates of nodes by those specifications and running time. In addition, the approach to specify and guarantee the reliability as the targeting system reliability is enough assumed if the approximate churn rates of nodes can be estimated. The churn rate is also expressed in terms of the reliability of a given node, with R denoting the reliability of the node, and the churn rate is expressed as 1 - R.

• The number of destinations for each delivery cycle The proposed method varies the degree of robustness enhancement for each delivery cycle, based on the number of the respective destinations, because the robustness enhancement of nodes varies with this number. For example, if a node assigned to a cycle group with many destinations churns, the system reliability is significantly damages. Therefore, in this paper, we assume that nodes know the approximate number of destinations for each cycle, and nodes of cycle groups with more destinations have higher robustness in the proposed method.

3.2 Determination of Replication Nodes

3.2.1 Non-Longest Cycle Group

In this paper, we call the group with the longest cycle at each specified time the "longest cycle group," and groups which do not have the longest cycle at that time "non-longest cycle groups." R_0 denotes the reliability of a node assigned to time t in one of the non-longest cycle groups, and R_u $(u = 1, 2, \dots)$ denotes the respective reliabilities of its replication node candidates, in order. If the original node sends replicated data to v candidate nodes, the reliability of the cycle group at time t is calculated by $1 - \prod_{u=0}^{v} (1 - R_u)$. In the existing method using distributed hashing, the longest cycle group at each specified time relays sensor data to the non-longest cycle groups. The respective reliability over $\sqrt{SR_i}$, because a churn in the longest cycle group is likely.

In the proposed method, the assigned nodes in the nonlongest cycle groups send replicated data to v candidate nodes at each specified time, according to the following equation:

$$1 - \prod_{u=0}^{v} (1 - R_u) \ge \sqrt{SR_t} \tag{1}$$

Considering the upper limitation of replication node candidates denoted by v, we assume that the delivery system determines v based on the number of nodes, estimated churn rates, or targeting system reliability. The large v causes the case to make all the nodes successors if the reliabilities of nodes are low. In addition, the targeting system reliability is possible not to be satisfied even if all the nodes become successors.

3.2.2 Longest Cycle Group

 C_i $(i = 1, 2, \dots, c)$ denotes selectable delivery cycles, and m_{it} $(i = 1, 2, \dots, c)$ denotes the number of destinations for each cycle group at time *t*. M_t denotes the number of destinations in the longest cycle group at time *t*, and the total number of destinations in the non-longest cycle groups is calculated by $\sum_{i=1}^{c} m_{it} - M_t$. If the reliability of the non-longest cycle groups is assumed to be approximately $\sqrt{SR_t}$, the expected value of the number of destinations in the non-longest cycle groups at time *t*, E_t , is calculated by the following equation:

$$E_t = \left(\sum_{i=1}^c m_{it} - M_t\right) \sqrt{SR_t} \tag{2}$$

 R_0 denotes the reliability of a node assigned to time t

in the longest cycle group, and R_u ($u = 1, 2, \cdots$) denotes the respective reliabilities of its replication node candidates, in order. If the original node sends replicated data to v candidate nodes, the reliability of the longest cycle group at time t is similar to that of the non-longest cycle groups, and calculated by $1 - \prod_{u=0}^{v} (1 - R_u)$. If the assigned node in the longest cycle group has not churned at time t, the expected value of the number of destinations in the whole system is denoted by $M_t + E_t$. In the proposed method, the assigned node in the longest cycle group sends replicated data to v candidate nodes at each specified time, according to the following equation:

$$\left(1 - \prod_{u=0}^{v} (1 - R_u)\right) \frac{M_t + E_t}{\sum_{i=1}^{c} m_{it}} \ge SR_t$$
(3)

4. Evaluation

In this study, we evaluated the method proposed in Sect. 3, through simulations. The simulator and all the methods were implemented by ourselves in Java programming language. Table 1 shows the simulations parameters, and those details are described below.

4.1 Simulation Environment

In the simulation environment, the number of nodes is $n = 2^7 = 128$, the number of sensor data streams is l = 1, and the number of destinations is m = 1000. The delivery cycles that destinations request are $C_i = i$ ($i = 1, \dots, 10$), and determined at random between 1 and 10. In this environment, the maximum of the least common multiple of delivery cycles is 2520, and thus the timetable for delivery is from time 0 to 2519. The scale of the unit time is variable, but this paper assumes 1 minute. The targeting system reliability is constant at all times, with $SR_t = 0.9$ ($t = 0, 1, \dots, 2519$) in Sects. 4.2 and 4.3.

In this simulation, we compared the proposed method with a method in which the number of replication nodes at

Table 1 Simulation parameters

Parameter	Value
The number of nodes <i>n</i>	$2^7 = 128$
The number of data streams <i>l</i>	1
The number of destinations m	1000
Delivery cycles C_i	$1, \cdots, 10$ (determined at random)
Simulation time t	$0, \cdots, 2519$
Targeting system reliability SR _t	0.9 (Sects. 4.2 and 4.3)
	$0.0, \cdots, 0.9$ (Sect. 4.4)
Basis nodal reliability Rbs	$0.1, \cdots, 1.0$
	(churn rate is expressed as $1 - R_{bs}$)
Nodal reliability model	Constant, bathtub-shaped, random
Comparison methods	No succs., 1 succ., 2 succs,
	4 succs., 8 succs, static, proposed
Simulation count	10
Evaluation items	Max. number of replication nodes,
	total number of replication nodes,
	min. of the system reliability,
	avg. of the system reliability



Fig. 3 The maximum number of instantaneous replication nodes on the constant scenario

all times is 0, 1, 2, 4, and 8; and with a method in which the number of replication nodes is determined at all times, in order to satisfy the targeting system reliability, based on the churn rates of nodes. Although the proposed method varies the number of replication nodes at each specified time, the second method employs a constant number of replication nodes at all times, as determined by the maximum value among all times; thus, we call this the "static method."

We calculated the number of replication nodes and the system reliability among the time of the least common multiple of selectable delivery cycles. The number of replication nodes relates to the cost of maintaining robustness. We executed this simulation 10 times for each method and environment, and calculated the averages of the results. Each environment has different basis nodal reliabilities and nodal reliability models. The basis nodal reliability is denoted as $R_{\rm bs}$, and its churn rate is expressed as $1 - R_{\rm bs}$. In addition, this paper uses three models called "constant scenario," "bathtub-shaped scenario," and "random scenario" to determine each nodal reliability. The details of the nodal reliability ity models are described in the next section.

4.2 Number of Replication Nodes

Figures 3, 4, and 5 show the maximum instantaneous number of replication nodes for each cycle group and time, based on the churn rate of nodes. Figure 3 shows the result when the nodal churn rate is constant at the value on the lateral axis (constant scenario). Figure 4 shows the result when the nodal churn rate is individually determined based on the bathtub-shaped model where the basis churn rate is the value on the lateral axis (bathtub-shaped scenario). Although each nodal reliability varies widely in the real world, this paper employs the disk failure rates shown by Backblaze [18] and individually determines the nodal year at random between 0 and 5. Figure 6 shows the used bathtub shape, and $R_{\rm bs}$ denotes the basis nodal reliability that relates to the basis churn rate. This paper assumes the simulation time is from time 0 to 2519 and the unit time is 1 minute. So, each nodal reliability has almost no changes during the simulation. Figure 5



Fig. 4 The maximum number of instantaneous replication nodes on the bathtub-shaped scenario



Fig.5 The maximum number of instantaneous replication nodes on the random scenario



Fig. 6 The used bathtub shape to determine the churn rate for each node

shows the result when the nodal churn rate is individually determined at random between 0 and 1 (random scenario). The longitudinal axis shows the maximum number of replication nodes for each cycle group and time. To take an ex-



Fig. 7 Total number of instantaneous replication nodes on the constant scenario



Fig. 8 Total number of instantaneous replication nodes on the bathtub-shaped scenario

ample, in this simulation environment, when the churn rate of all nodes is 0.9, in the constant scenario shown in Fig. 3, the maximum number of replication nodes in the unit time is 28. As the churn rate increases, the maximum number of replication nodes increases, in order to satisfy the targeting system reliability. The maximum number of replication nodes is least in the random scenario shown in Fig. 5, and the larger the difference between churn rates, the smaller the maximum number of replication nodes.

Figures 7, 8, and 9 show the total number of replication nodes, based on the nodal churn rate. The environment is identical to that of Figs. 3, 4, and 5, with the longitudinal axis shows the total number of replication nodes at time 0 to 2519.

Similar to the results for the maximum number of replication nodes, the higher the churn rate, the larger the total number of replication nodes. In addition, the static method, which determines replication nodes based on the churn rate, shows results similar to those of the proposed method, with respect to the maximum number of replication nodes. However, the total number of replication nodes in the proposed method is less than in the static method, especially as seen



Fig.9 Total number of instantaneous replication nodes on the random scenario



Fig. 10 Cumulative relative frequency of the number of instantaneous replication nodes

in Figs. 8 and 9, because the proposed method changes the number of replication nodes each time, based on situations such as the churn rate. In the environment where the nodal churn rates differ, the number of replication nodes also differ, to satisfy the targeting system reliability at each time. Therefore, the proposed method, which changes the number of replication nodes at each time, reduces the costs related to unnecessary replication nodes.

Figure 10 shows the cumulative relative frequency from time 0 to 2519 against the number of instantaneous replication nodes in the unit time. The lateral axis shows the number of replication nodes in the unit time, and the longitudinal axis shows the cumulative relative frequency of times under the number of replication nodes shown on the lateral axis. Figure 10 shows the maximum number of replication nodes at each time, in the static method and proposed method. The churn rates of nodes are determined at random between 0 and 1.

In Fig. 10, the maximum number of replication nodes in the static method is always 11. On the other hand, the maximum number of replication nodes in the proposed method is less than 4 roughly 60% of the times. Moreover, the maximum number of replication nodes in the proposed method is less than 8 at most times.



Fig. 11 The minimum of the instantaneous system reliability on the constant scenario



Fig. 12 The minimum of the instantaneous system reliability on the bathtub-shaped scenario



Fig. 13 The minimum of the instantaneous system reliability on the random scenario

4.3 System Reliability

Figures 11, 12, and 13 show the minimum of the instantaneous system reliability. The instantaneous system reliability shows the rate of destinations successfully receiving



Fig.14 The average of the instantaneous system reliability on the constant scenario



Fig. 15 The average of the instantaneous system reliability on the bathtub-shaped scenario

data at each time. Figure 11 shows the result when the nodal churn rate is constant at the value on the lateral axis. Figure 12 shows the result when the modal churn rate is individually determined based on the bathtub-shaped model similar to Fig. 4. Figure 13 shows the result when the nodal churn rate is individually determined at random between 0 and 1. The longitudinal axis shows the minimum of the instantaneous system reliability. Figures 14, 15, and 16 show the average of the instantaneous system reliability by the nodal churn rate in the same environment. The longitudinal axis shows the average of the instantaneous system reliability.

In the proposed method, as shown in Figs. 11, 12, and 13, even the minimum of the system reliability satisfied the targeting system reliability, $SR_t = 0.9$. The static method also satisfied the targeting system reliability; however, the difference from the targeting system reliability is larger than in the proposed method, especially in the environment where the churn rates differ, such as in Figs. 12 and 13. The results in Figs. 7, 8, and 9 suggest that the proposed method avoids sending replicated data to unnecessary nodes



Fig. 16 The average of the instantaneous system reliability on the random scenario



Fig. 17 The maximum number of instantaneous replication nodes on the random scenario

in satisfying the targeting system reliability. In addition, the proposed method can dynamically adapt to the change of churn situations by updating the number of replication nodes based on nodal information such as the operating time.

4.4 Results by the Targeting System Reliability

Figures 17, 18, 19, and 20 show each result on the random scenario when the targeting system reliability, SR_t , is constant at the value on the lateral axis ($t = 0, 1, \dots, 2519$). The results except the static method and proposed are the same to the shown results in Sects. 4.2 and 4.3 because SR_t affects the number of successors only in the static method and proposed method.

Similar to the results in Sect. 4.2, the maximum number of replication nodes in Fig. 17 is the same for the static method and proposed method, however, the total number of replication nodes of the proposed method in Fig. 18 shows less values than that of the static method. The difference between the static method and proposed method is increased by SR_i . Also similar to the results in Sect. 4.3, even the minimum of the system reliability in Fig. 19 satisfied the targeting system reliability, SR_i , in all the results. In addition, the difference from SR_i in the proposed method is less than that of the static method in both of Figs. 19 and 20. Es-



 $\label{eq:Fig.18} Fig.18 \quad \mbox{Total number of instantaneous replication nodes on the random scenario}$



Fig. 19 The minimum of the instantaneous system reliability on the random scenario



Fig. 20 The average of the instantaneous system reliability on the random scenario

pecially, the average of the system reliability of the static method in Fig. 20 reaches over 90% even if SR_t is 0.1. This result means that the static method sends replicated data to

many unnecessary nodes compared to the proposed method.

5. Related Work

A variety of replication schemes, such as path replication [19], have been proposed for unstructured P2P networks, where nodes search for content by forwarding queries to publishing nodes via neighboring links; and the path replication schemes replicate the replied content on the nodes between the publishing and requesting nodes. Related to the path replication schemes, a number of methods have been proposed based on specific factors such as the number of queries, the probability to put replicas, churn situations, and so on [20]–[22]. However, we assume a structured P2P network such as Chord. Although structured P2P networks require more frequent link maintenance than unstructured P2P networks, they enable higher efficiency and accuracy in content searches.

Replication schemes have also been proposed for structured P2P networks, to increase the efficiency of replica maintenance and searches. Scalaris, for example, an Erlang implementation of a distributed key/value store [23], uses replication for data availability and majority-based distributed transactions for data consistency. Plover is a proactive low-overhead file replication scheme with replication among physically proximate nodes based on their available capacities [24]. Here, the physically proximate nodes are grouped in clusters, each of which has a supernode with high capacity and rapid connections. In RelaxDHT, nodes are divided into data blocks [25], with each block having a root node that manages the metadata of replicas on other nodes in their own different data blocks. However, unlike the proposed system, none of these schemes guarantee the specified reachability under churn situations [24], [26], [27]. In addition, existing schemes require the optional function and maintenance from overlay.

6. Conclusion

In this paper, we proposed a method for constructing a scalable sensor data stream delivery system that guarantees the specified QoS of the delivery (i.e., total reachability to destinations) even when the delivery server resources are in a heterogeneous churn situation. Through simulations, we confirmed that the proposed method can guarantee the required reachability, while avoiding any increase in unnecessary resource assignment costs.

In the future, we will study a related technique applicable to environments where the number of destinations for each delivery cycle tends to be large and have large differences among delivery cycles. In addition, we plan to evaluate our proposed method in more realistic environments where the estimated churn rates or reliabilities of nodes have large errors from those true values.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 16K16059, 17K00146, Hoso Bunka Foundation, and Kayamori Foundation of Informational Science Advancement, Japan.

References

- T. Kawakami, Y. Ishi, T. Yoshihisa, and Y. Teranishi, "A P2P-based sensor data stream delivery method to accommodate heterogeneous cycles," Journal of Information Processing (JIP), vol.22, no.3, pp.455–463, July 2014.
- [2] X. Zhang, J. Liu, B. Li, and Y.-S.P. Yum, "CoolStreaming/DONet: A data-driven overlay network for peer-to-peer live media streaming," Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005), pp.2102–2111, March 2005.
- [3] X. Liao, H. Jin, Y. Liu, L.M. Ni, and D. Deng, "Anysee: Peer-to-peer live streaming," Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM 2006), pp.1–10, April 2006.
- [4] T. Moscibroda and R. Rejaie, "PRIME: Peer-to-peer receiverdriven mesh-based streaming," Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM 2007), pp.1415–1423, May 2007.
- [5] L. Yu, X. Liao, H. Jin, and W. Jiang, "Integrated buffering schemes for P2P VoD services," Peer-to-Peer Networking and Applications, vol.4, no.1, pp.63–74, 2011.
- [6] S. Sakashita, T. Yoshihisa, T. Hara, and S. Nishio, "A data reception method to reduce interruption time in P2P streaming environments," Proceedings of the 13th International Conference on Network-Based Information Systems (NBiS 2010), pp.166–172, Sept. 2010.
- [7] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2002), pp.205–217, Aug. 2002.
- [8] D.A. Tran, K.A. Hua, and T. Do, "ZIGZAG: An efficient peerto-peer scheme for media streaming," Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003), pp.1283–1292, March 2003.
- [9] X. Jin, W.-P.K. Yiu, S.-H.G. Chan, and Y. Wang, "On maximizing tree bandwidth for topology-aware peer-to-peer streaming," IEEE Trans. Multimedia, vol.9, no.8, pp.1580–1592, Dec. 2007.
- [10] K. Silawarawet and N. Nupairoj, "Locality-aware clustering application level multicast for live streaming services on the Internet," Journal of Information Science and Engineering, vol.27, no.1, pp.319– 336, 2011.
- [11] T.A. Le and H. Nguyen, "Application-aware cost function and its performance evaluation over scalable video conferencing services on heterogeneous networks," Proceedings of the IEEE Wireless Commun. and Networking Conference: Mobile and Wireless Networks (WCNC 2012 Track 3 Mobile and Wireless), pp.2185–2190, April 2012.
- [12] V. Gulisano, R. Jiménez-Peris, M. Patiño-Martínez, C. Soriente, and P. Valduriez, "StreamCloud: An elastic and scalable data streaming system," IEEE Trans. Parallel Distrib. Syst., vol.23, no.12, pp.2351–2365, Dec. 2012.
- [13] T. Kawakami, Y. Ishi, T. Yoshihisa, and Y. Teranishi, "A sensor data stream delivery method to accommodate heterogeneous cycles on cloud," IEICE Trans. Commun., vol.E99-B, no.6, pp.1331–1340, June 2016.
- [14] T. Kawakami, Y. Ishi, T. Yoshihisa, and Y. Teranishi, "A churn resilience technique on P2P sensor data stream delivery system using

distributed hashing," International Journal of Computing and Informatics (Informatica), vol.39, no.4, pp.355–363, Dec. 2015.

- [15] T. Kawakami, Y. Ishi, T. Yoshihisa, and Y. Teranishi, "A node replication method to guarantee reachability for P2P sensor data stream delivery system on heterogeneous churn situations," Proceedings of the 2nd IEEE International Workshop on Big Data Management for the Internet of Things (BIoT 2015) in Conjunction with the 39th Annual International Computer, Software and Applications Conference (COMPSAC 2015), pp.529–534, July 2015.
- [16] I. Stoica, R. Morris, D. Liben-Nowell, D.R. Karger, M.F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup protocol for internet applications," IEEE/ACM Trans. Netw., vol.11, no.1, pp.17–32, Feb. 2003.
- [17] M. Xie, Y. Tang, and T.N. Goh, "A modified Weibull extension with bathtub-shaped failure rate function," Reliability Engineering & System Safety, vol.76, no.3, pp.279–285, June 2002.
- [18] Backblaze Blog, "How long do disk drives last?," available at http://www.backblaze.com/how-long-do-disk-drives-last.html (accessed Aug. 1, 2018).
- [19] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," Proceedings of the 16th ACM International Conference on Supercomputing (ICS 2002), New York, New York, USA, pp.84–95, June 2002.
- [20] V. Gopalakrishnan, B. Silaghi, B. Bhattacharjee, and P. Keleher, "Adaptive replication in peer-to-peer systems," Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS 2004), pp.360–369, March 2004.
- [21] H. Yamamoto, D. Maruta, and Y. Oie, "Replication methods for load balancing on distributed storages in P2P networks," IEICE Trans. Inf. & Syst., vol.E89-D, no.1, pp.171–180, Jan. 2006.
- [22] K.P.N. Puttaswamy, A. Sala, and B.Y. Zhao, "Searching for rare objects using index replication," Proceedings of the 27th IEEE International Conference on Computer Communications (INFOCOM 2008), pp.1723–1731, April 2008.
- [23] T. Schütt, F. Schintke, and A. Reinefeld, "Scalaris: Reliable transactional P2P key/value store," Proceedings of the 7th ACM SIGPLAN Workshop on ERLANG (ERLANG 2008), pp.41–48, Sept. 2008.
- [24] H. Shen and Y. Zhu, "A proactive low-overhead file replication scheme for structured P2P content delivery networks," Journal of Parallel and Distributed Computing, vol.69, no.5, pp.429–440, May 2009.
- [25] S. Legtchenko, S. Monnet, P. Sens, and G. Muller, "RelaxDHT: A churn-resilient replication strategy for peer-to-peer distributed hash-tables," ACM Transactions on Autonomous and Adaptive Systems (TAAS), vol.7, no.2, Article 28, July 2012.
- [26] A. Pace, V. Quéma, and V. Schiavoni, "Exploiting node connection regularity for DHT replication," Proceedings of the 30th IEEE International Symposium on Reliable Distributed Systems (SRDS 2011), pp.111–120, Oct. 2011.
- [27] M. Ferreira, J. Leitão, and L. Rodrigues, "Thicket: A protocol for building and maintaining multiple trees in a P2P overlay," Proceedings of the 29th IEEE Symposium on Reliable Distributed Systems (SRDS 2010), pp.293–302, Oct. 2010.



Tomoya Kawakami received his B.E. degree from Kinki University in 2005 and his M.I. and Ph.D. degrees from Osaka University in 2007 and 2013, respectively. From 2007 to March 2013 and from July 2014 to March 2015, he was a specially appointed researcher at Osaka University. From April 2013 to June 2014, he was a Ph.D. researcher at Kobe University. Since April 2015, he has been a assistant professor at Nara Institute of Science and Technology. His research interests include dis-

tributed computing, rule-based systems, and stream data processing. He is a member of the IPSJ and IEEE.



Tomoki Yoshihisa received the Bachelor's, Master's, and Doctor's degrees from Osaka University, Osaka, Japan, in 2002, 2003, 2005, respectively. Since 2005 to 2007, he was a research associate at Kyoto University. In January 2008, he joined the Cybermedia Center, Osaka University as an assistant professor and in March 2009, he became an associate professor. From April 2008 to August 2008, he was a visiting researcher at University of California, Irvine. His research interests include video-

ondemand, broadcasting systems, and webcasts. He is a member of the IPSJ, IEICE, and IEEE.



Yuuichi Teranishi received his M.E. and Ph.D. degrees from Osaka University, Japan, in 1995 and 2004, respectively. From 1995 to 2004, he was engaged Nippon Telegraph and Telephone Corporation (NTT). From 2005 to 2007, he was a Lecturer of Cybermedia Center, Osaka University. From 2007 to 2011, He was an associate professor of Graduate School of Information Science and Technology, Osaka University. Since August 2011, He has been a research manager and project manager of Na-

tional Institute of Information and Communications Technology (NICT). He received IPSJ Best Paper Award in 2011. His research interests include technologies for distributed network systems and applications. He is a member of the IPSJ and IEEE.