

# A Sensor Data Stream Collection Scheme Considering Phase Differences for Load Balancing

Tomoya Kawakami, Graduate School of Engineering, University of Fukui, Japan

Tomoki Yoshihisa, Cybermedia Center, Osaka University, Japan

Yuuichi Teranishi, National Institute of Information and Communications Technology, Japan

## ABSTRACT

In the internet of things (IoT), various devices (things) including sensors generate data and publish them via the internet. The authors define continuous sensor data with difference cycles as a sensor data stream and have proposed methods to collect distributed sensor data streams. In this paper, the authors describe a skip graph-based collection scheme for sensor data streams considering phase differences. In the proposed scheme considering phase differences, the collection time is balanced within each collection cycle by the phase differences, and the probability of load concentration to the specific time or node is decreased. The simulation results show that the proposed scheme can equalize the loads of nodes even if the distribution of collection cycles is not uniform.

## KEYWORDS

Collection Cycle, DAT, Data Aggregation, Distributed Aggregation Tree, Load Distribution, Phase Shifting, Skip Graphs, Structured Overlay Network, TBPS, Topic-Based Pub/Sub

## INTRODUCTION

The Internet of Things (IoT) (Hodges et al., 2013) has attracted greater interest and attention with the spread of network-connected small devices such as sensors, smartphones, and wearable devices. In the data science field, stream data generated from IoT devices are analyzed to get various information. A larger amount of data can lead to higher-quality information such a faster stream data collection is one of the main techniques in the data science field and various schemes have been proposed. To enable IoT applications for data collection, pub/sub messaging (Eugster et al., 2003) is considered to be a promising event delivery method that can achieve the asynchronous dissemination and collection of information in real-time in a loosely-coupled environment. For example, the sensor devices correspond to publishers, and the IoT application corresponds to a subscriber. Topic-Based Pub/Sub (TBPS) protocols are already widely utilized by many IoT applications (Teranishi et al., 2015; Teranishi et al., 2017). These systems have a broker server for managing topics. The broker gathers all the published messages and forwards them to the corresponding subscribers.

DOI: 10.4018/IJMCMC.20210401.oa1

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

In IoT and Big Data applications, collecting all of the raw (unfiltered) sensor data is important for conducting various forms of analysis (Bessis et al., 2014). In this case, the larger the number of sensors treated on the application for analysis, the larger the number of messages that need to be received per time unit on the broker and subscribers in TBPS. For example, when the publishers correspond to a certain kind of sensor which publishes sensor data every 10 seconds and the number of target sensors in an application is 10,000, the broker must receive 1,000 messages per second on average. Thus, the number of messages tends to explode on the broker and the subscribers in IoT and Big Data applications. In general, the number of sent/received messages per unit of time affects the network process load because tasks such as adding/removing headers and serializing/deserializing payloads are required for each message. Therefore, even though the size of each sensor data is small, the increase in the number of publishers can cause network process overloads on the broker and subscribers. This leads to the loss of data or unusual increases in delivery latency, problems which have harm on IoT and Big Data applications.

Many existing studies tackle the problem of scalability in TBPS systems. The approach of these studies is based on distributed brokers, in which brokers are run as peers in a peer-to-peer system. The brokers construct an overlay network among themselves. For example, there are distributed hash table (DHT)-based approaches (Castro et al., 2002; Ratnasamy et al., 2001), hybrid overlay approaches (Rahimian et al., 2011), and Skip Graph-based (Aspnes et al., 2007; Shao et al., 2015; Banno & Fujio et al., 2015; Banno et al., 2020) approaches (Banno & Takeuchi et al., 2015; Teranishi et al., 2015). These approaches can keep the number of connections that each broker needs to accept small by multi-hop message forwarding on overlays. However, these existing methods aim to deliver messages from one publisher to multiple subscribers in a scalable manner. Thus, they are unable to avoid network process overloads caused by the collection, such as when messages are received from a large number of publishers. In addition, the existing techniques do not assume the different intervals at the same time to periodically collect data from the publishers.

Therefore, we define continuous sensor data with different intervals (cycles) as a sensor data stream and have proposed collection methods for distributed sensor data streams as a topic-based pub/sub (TBPS) system (Teranishi et al., 2017; Kawakami et al., 2017). Especially in Teranishi et al., 2017, we have proposed a message forwarding scheme on overlays called “Collective Store and Forwarding.” The scheme can reduce loads of network processes dramatically when there are a large number of publishers (nodes), maintaining the delivery time constraints given for the messages. In addition, we have also proposed a flexible tree construction method called “Adaptive Data Collection Tree” on Chord# (Schütt et al., 2008) that can adjust the maximum load of network processes on distributed brokers to avoid overloads caused by Collective Store and Forwarding. Moreover, we have proposed an expanded method assigning phase differences to balance the collection time among the same or specific collection cycle nodes (Kawakami et al., 2018; Kawakami et al., 2019). We call this novel approach “phase shifting (PS).” The PS approach enables the SG method to decrease the probability of load concentration to the specific time or node. Assigning phase differences at random to the nodes, the collection times are distributed even if there are the same collection cycle nodes. We have evaluated our proposed methods in simulation. Our experiment results show that our proposed method can reduce loads of nodes and realize highly scalable systems to periodically collect distributed sensor data. The scalability of the data collection systems is significantly important to accommodate a huge number of objects and encourage the growth of the data science field.

In the following, the problems addressed are defined in the second section. The data collection scheme considering phase differences is described in the third section. We describe the discussion and related work in the fourth and fifth sections, respectively. Finally, the conclusion of the chapter is presented in the sixth section.

## PROBLEMS ADDRESSED

### Assumed Environment

The purpose of this study is to disperse the communication load in the sensor stream collections that have different collection cycles. The source nodes have sensors to gain sensor data periodically. The source nodes and collection node (sink node) of those sensor data construct P2P networks. The sink node searches source nodes and requires a sensor data stream with those collection cycles in the P2P network. Upon reception of the query from the sink node, the source node starts to deliver the sensor data stream via other nodes in the P2P network. The intermediate nodes relay the sensor data stream to the sink node based on their routing tables.

### Input Setting

The source nodes are denoted as  $N_i$  ( $i = 1, \dots, n$ ), and the sink node of sensor data is denoted as  $S$ . In addition, the collection cycle of  $N_i$  is denoted as  $C_i$ .

In Figure 1, each node indicates source nodes or sink node, and the branches indicate collection paths for the sensor data streams. Concretely, they indicate communication links in an application layer. The branches are indicated by dotted lines because there is a possibility that the branches may not collect a sensor data stream depending on the collection method. The sink node  $S$  is at the top and the four source nodes  $N_1, \dots, N_4$  ( $n = 4$ ) are at the bottom. The figure in the vicinity of each source node indicates the collection cycle, and  $C_1 = 1$ ,  $C_2 = 2$ ,  $C_3 = 2$ , and  $C_4 = 3$ . This corresponds to the case where a live camera acquires an image once every second, and  $N_1$  records the image once every second,  $N_2$  and  $N_3$  record the image once every two seconds, and  $N_4$  records the image once every three seconds, for example. Table 1 shows the collection cycle of each source node and the sensor data to be received in the example in Figure 1. The purpose of this study is to disperse the communication load in the sensor stream collections that have different collection cycles. The source nodes have sensors to gain sensor data periodically. The source nodes and collection node (sink node) of those sensor data construct P2P networks. The sink node searches source nodes and requires a sensor data stream with those collection cycles in the P2P network. Upon reception of the query from the sink node, the source node starts to deliver the sensor data stream via other nodes in the P2P network. The intermediate nodes relay the sensor data stream to the sink node based on their routing tables.

### Definition of a Load

The communication load of the source nodes and sink node is given as the total of the load due to the reception of the sensor data stream and the load due to the transmission. The communication load due to the reception is referred to as the reception load, the reception load of  $N_i$  is  $I_i$  and the reception load of  $S$  is  $I_0$ . The communication load due to the transmission is referred to as the transmission load, the transmission load of  $N_i$  is  $O_i$  and the transmission load of  $S$  is  $O_0$ .

In many cases, the reception load and the transmission load are proportional to the number of sensor data pieces per unit hour of the sensor data stream to be sent and received. The number of pieces of sensor data per unit hour of the sensor data stream that is to be delivered by  $N_p$  to  $N_q$  ( $q \neq p$ ;  $p, q = 1, \dots, n$ ) is  $R(p, q)$ , and the number delivered by  $S$  to  $N_q$  is  $R(0, q)$ .

## SKIP GRAPH-BASED COLLECTION SYSTEM CONSIDERING PHASE DIFFERENCES

### Skip Graphs

In this paper, we assume the overlay network for the skip graph-based TBPS such as Banno & Takeuchi et al., 2015.

Figure 1. An example of input settings

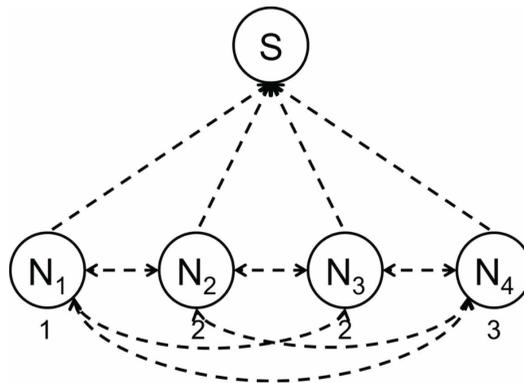


Table 1. An example of the sensor data collection

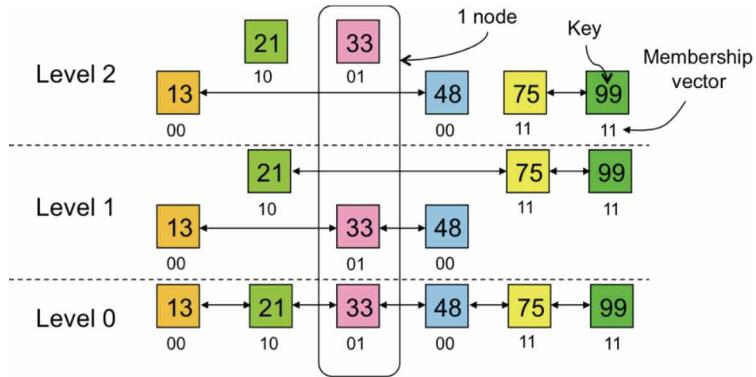
Time	N <sub>1</sub> (Cycle = 1)	N <sub>2</sub> (Cycle = 2)	N <sub>3</sub> (Cycle = 2)	N <sub>4</sub> (Cycle = 3)
0	*	*	*	*
1	*			
2	*	*	*	
3	*			*
4	*	*	*	
5	*			
6	*	*	*	*
7	*			
...	...	...	...	...

Skip graphs are overlay networks that skip list are applied in the P2P model (Aspnes et al., 2007). Figure 2 shows the structure of a skip graph. In Figure 2, squares show entries of routing tables on peers (nodes), and the number inside each square shows a key of the peer. The peers are sorted in ascending order by those keys, and bidirectional links are created among the peers. The numbers below entries are called “membership vector.” The membership vector is an integral value and assigned to each peer when the peer joins. Each peer creates links to other peers on the multiple levels based on the membership vector. In skip graphs, queries are forwarded by the higher-level links to other peers when a single key and its assigned peer is searched. This is because of the higher-level links can efficiently reach the searched key with fewer hops than the lower level links. In the case of range queries that specify the beginning and end of keys to be searched, the queries are forwarded to the peer whose key is within the range, or less than the end of the range. The number of hops to key search is represented to  $O(\log n)$  when  $n$  is denoted as the number of peers. In addition, the average number of links on each peer is represented to  $\log n$ .

### Phase Differences

Currently, we have proposed a large-scale data collection schema for distributed TBPS (Teranishi et al., 2017). In Teranishi et al., 2017, we employ “Collective Store and Forwarding,” which stores and merges multiple small size messages into one large message along with a multi-hop tree structure

Figure 2. A structure of a skip graph



on the structured overlay for TBPS, taking into account the delivery time constraints. This makes it possible to reduce the overhead of network processes even when a large number of sensor data is published asynchronously. In addition, we have proposed a collection system considering phase differences (Kawakami et al., 2017; Kawakami et al., 2018; Kawakami et al., 2019). In the proposed method, the phase difference of the source node  $N_i$  is denoted as  $d_i$  ( $0 \leq d_i < C_i$ ). In this case, the collection time is represented to  $C_i p + d_i$  ( $p = 0, 1, 2, \dots$ ). Table 2 shows the time to collect data in the case of Figure 1 where the collection cycle of each source node is 1, 2, or 3. By considering phase differences like Table 2, the collection time is balanced within each collection cycle, and the probability of load concentration to the specific time or node is decreased. Each node sends sensor data at the time base on his collection cycle and phase difference, and other nodes relay the sensor data to the sink node. In this paper, we call considering phase differences “phase shifting (PS).” Figures 3 and 4 show an example of the data forwarding paths on skip graphs without phase shifting (PS) and with PS, respectively.

## EVALUATION

In this section, we describe the evaluation of the proposed skip graph-based method with phase shifting (PS) by simulation.

### Collection Target Nodes

Table 3 shows the simulation environments. We evaluate our proposed system in two environments by the combination of collection cycles. The number of the sink node is one, and the collection cycle

Table 2. An example of the collection time considering phase differences

Cycle	Phase Diff.	Collect Time
1	0	0, 1, 2, 3, 4, ...
2	0	0, 2, 4, 6, 8, ...
	1	1, 3, 5, 7, 9, ...
3	0	0, 3, 6, 9, 12, ...
	1	1, 4, 7, 10, 13, ...
	2	2, 5, 8, 11, 14, ...

Figure 3. An example of the skip graph-based method without PS

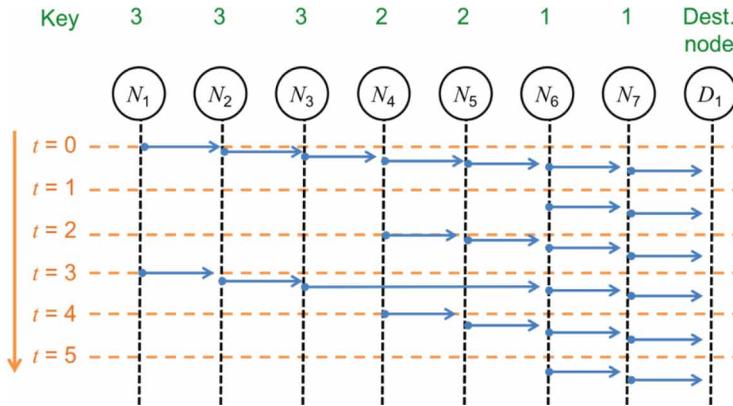


Figure 4. An example of the skip graph-based method with PS

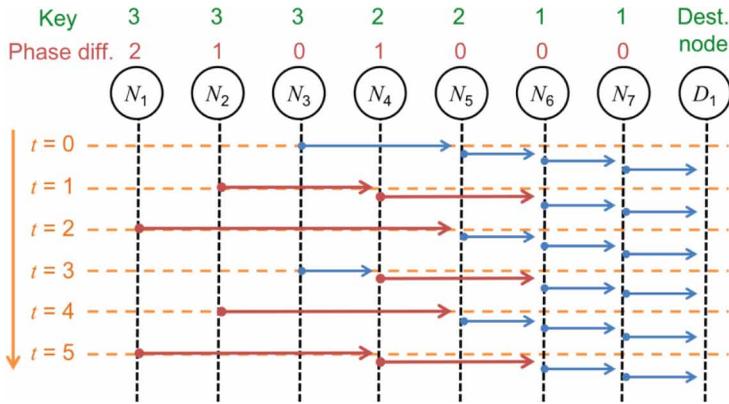


Table 3. Simulation environments

Env.	No. of Nodes	Cycles
1	1000	1, 2, 3
2	1000	1, 2, ..., 10

$C_i$  and phase difference  $d_i$  of each node is determined at random. In the simulations, we measure the number of nodes targeted to collect data from time 0 to 99 and compare the results with the case of not considering phase differences.

Figures 5 and 6 show the results of the number of nodes targeted to collect data from time 0 to 99. The horizontal axis shows the time, and the vertical axis shows the number of the targeted nodes at each time. In the simulation environment 1 shown by Figure 5, the case of not considering phase differences collects data from all 1000 nodes at time 0, 6, 12, ..., 96. This is because the collection cycle is 1, 2, or 3, and the lowest common multiple is 6. At other time in the case of not considering phase differences, the number of the nodes is extremely and constantly increase/decrease. On the other hand, the collection time is shifted by the phase differences in our proposed system, and the number of the nodes is probabilistically equalized for each time if the phase difference of each node

Figure 5. The number of the related nodes in Environment 1

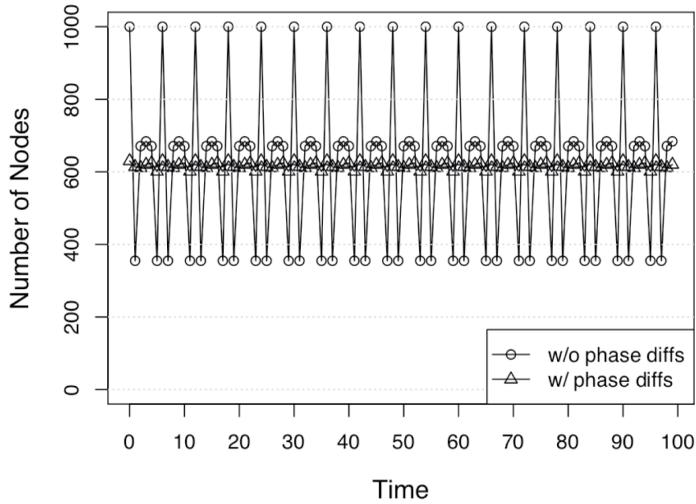
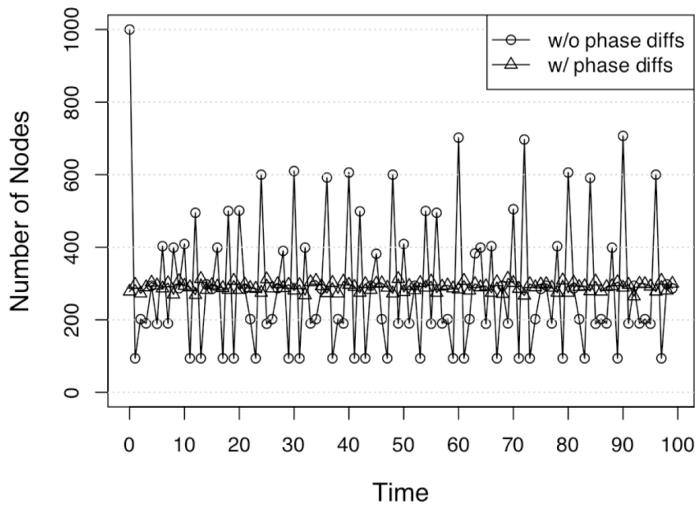


Figure 6. The number of the related nodes in Environment 2



is determined at random. Therefore, the probability of load concentration is decreased. Also in the simulation environment 2 shown by Figure 6, our proposed system achieves high balancing similar to the results in the simulation environment 1 while the case of not considering phase differences changes the number of the nodes complexly by the combination of cycles from 1 to 10.

### Communication Loads and Hops

In the simulation environments of our previous work, the collection cycle of each source node denoted as  $C_i$  is determined at random between 1 and 10. The selectable cycles are assumed limited by the practical systems, however, the distribution of the selected cycles is not uniform in the real world. In this paper, therefore, we evaluate our proposed method in the different distributions of collection cycles. The employed distributions are the normal (Gaussian) distribution and exponential distribution. To determine the integer cycle between 1 and 10, the normal distribution has 5.5 as an average and

1.5 as a variance. In addition, the exponential distribution determines the integer cycle for each node based on its cumulative distribution function (CDF),  $10(1 - e^{-x})$ , while  $x$  is determined between 0.0 and 5.0 at random. For other parameters, the simulation time denoted as  $t$  is from 0 to 2519, which length is the least common multiple of the selectable cycles. In addition, this simulation has no communication delays among nodes although there are various communication delays in the real world. As comparison methods, we compare the proposed method with the skip graph-based method without PS shown in Figure 4, the method in which all source nodes send data to the destination node directly (Source Direct, SD), and the method in which all source nodes send data to the next node for the destination node (Daisy Chain, DC). Figures 7 and 8 show an example of SD and DC with PS, respectively.

Figure 7. An example of the Server Direct (SD) method

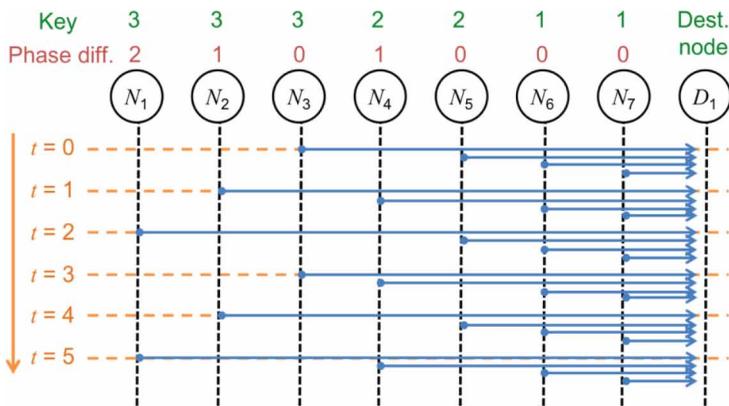
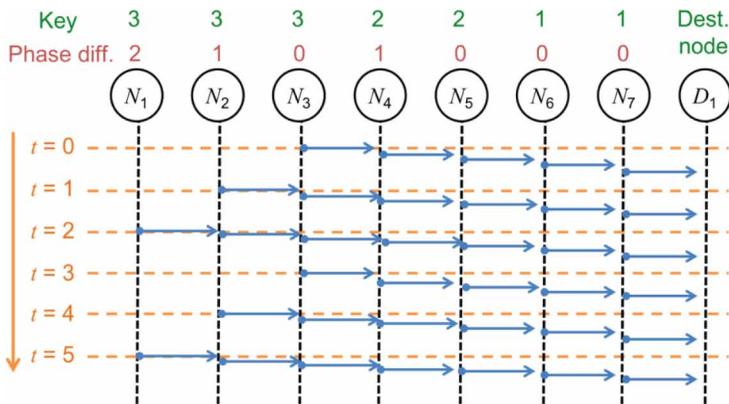


Figure 8. An example of the Daisy Chain (DC) method



Figures 9 and 10 show the results for the maximum instantaneous load and total loads of nodes by the number of nodes, respectively. The lateral axis shows the number of nodes, and the allowable number of stream aggregation is under 11. In all the distributions from Figure 9, the proposed method, skip graphs (SG) with PS, has a lower instantaneous load compared to SD-based methods where the destination node receives data directly from the source nodes. Although the larger the allowable number of stream aggregation in DC-based methods, the smaller the number of transmission and reception.

Figure 9. The maximum instantaneous load by the number of nodes

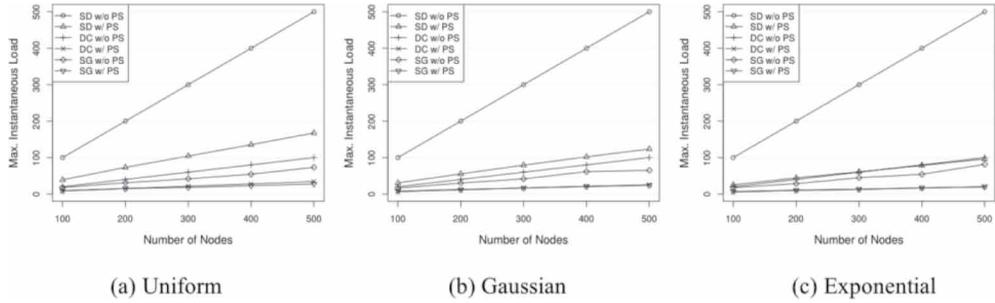
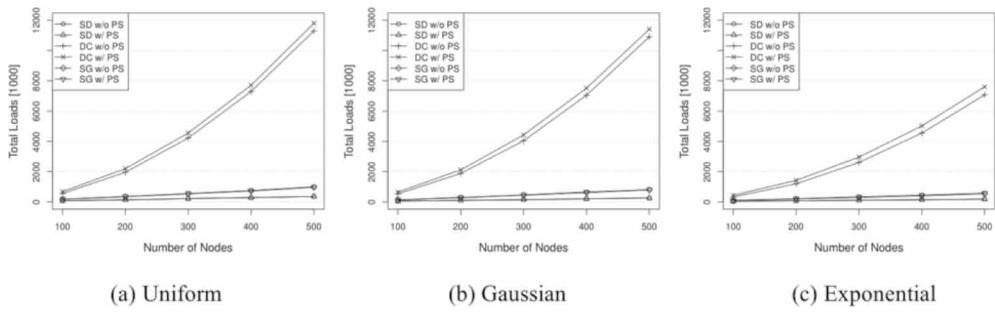


Figure 10. The total loads by the number of nodes



In this simulation environment, however, the proposed method has a lower instantaneous load than the results of DC-based methods. In addition, the proposed method has a lower instantaneous load compared to SG without PS because each node has different timing of transmission and reception by its phase difference even if another node is configured the same collection cycle. In Figure 10, on the other hand, SD-based methods have the lowest total loads. However, the proposed method has lower total loads compared to DC-based methods in this simulation environment. In addition, the total loads are the lowest in the exponential distribution because longer cycles have higher probabilities to be selected.

Similar to the results for the maximum instantaneous load and total loads of nodes, Figures 11 and 12 show the results for the average number and the maximum number of hops by the number of nodes under 11 streams aggregation, respectively. In Figures 11 and 12, SD-based methods have only one hop as the average number and maximum number although those instantaneous loads described in Figure 9 are high. The proposed method has  $\log n$  as the average number of hops while  $n$  is denoted as the number of nodes and DC-based methods are affected linearly by  $n$ .

Figures 13 and 14 show the results for the maximum instantaneous load and total loads of nodes by the allowable number of stream aggregation, respectively. The allowable number of stream aggregation is the value on the lateral axis, and the number of nodes is 200. SD-based methods have a constant value as the maximum instantaneous load which is not affected by the allowable number of stream aggregation because the source nodes send data to the destination node directly. In Figures 13 and 14, most of the results decrease by the increase of the allowable number of stream aggregation. The proposed method, SG with PS, has lower results for both of the maximum instantaneous load and total loads even in the realistic situation, 4<sup>1</sup> streams aggregation, compared to DC-based methods that require many streams aggregation to reduce those loads. In addition, the average number and

Figure 11. The average hops by the number of nodes

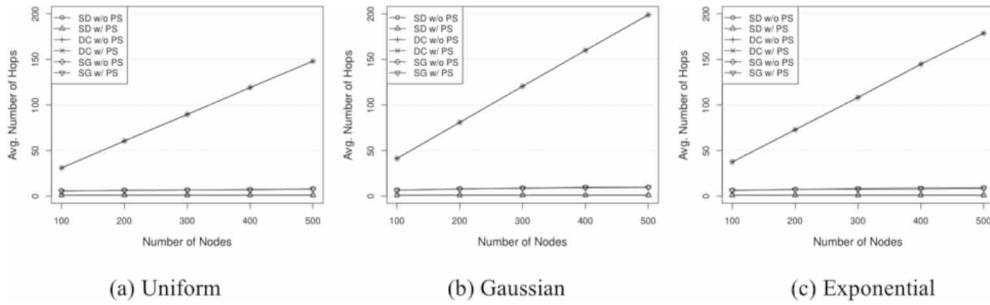


Figure 12. The maximum hops by the number of nodes

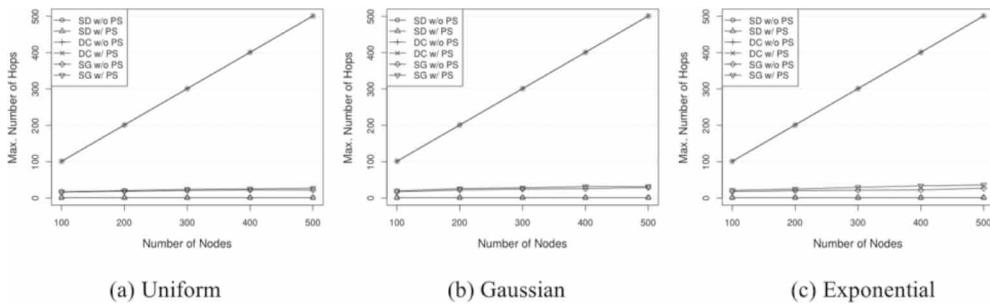


Figure 13. The maximum instantaneous load by the allowable number of stream aggregation

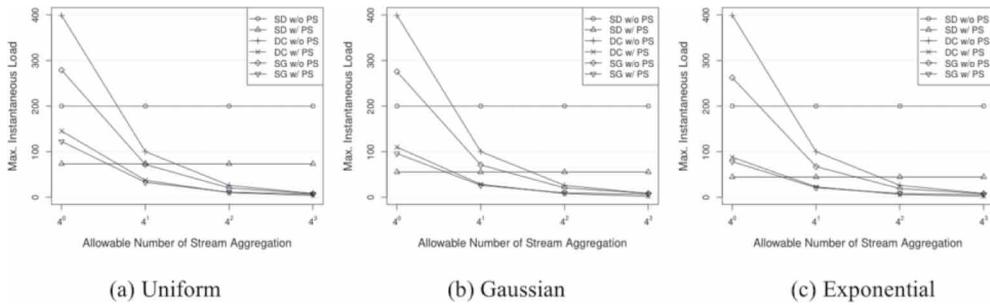
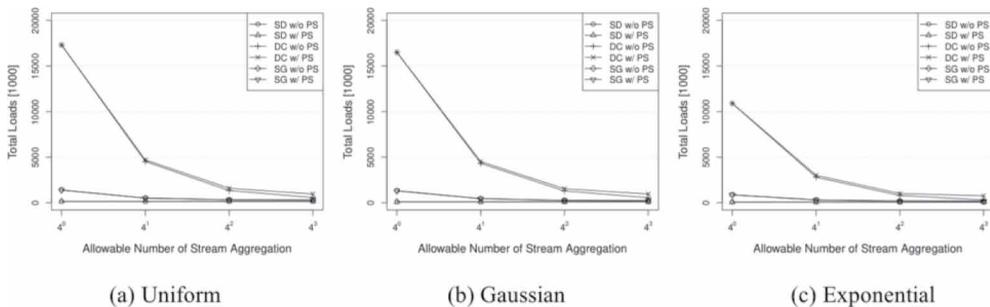


Figure 14. The total loads by the allowable number of stream aggregation



the maximum number of hops are the same as the results of 200 nodes in Figures 11 and 12 because they are not affected by the allowable number of stream aggregation.

## DISCUSSION

We described the data collection scheme with the approach of phase shifting (PS) in the third section. Our experiment results show that our proposed method can reduce loads of nodes and realize highly scalable systems to periodically collect distributed sensor data.

As the limitations of our current study, we assume the pieces of data are not so different from each other. In the real world, however, various types of data are published at the same time such as texts, images, and audio. Those pieces of data have different sizes and loads to be processed. We can clear this limitation by considering not only the number of data pieces (or transmission/reception) but also the types of them. Similar to the inconsideration of the data types, the inconsideration of nodes' performances is another limitation of our current study. We can clear this limitation by considering the nodes' performance such as processing power, memory size, and network environment. In addition, our current study has a limitation in the viewpoint of security or privacy. For example, private data are preferred to be sent to the subscriber via fewer nodes. Encryption of the data or communication is one of the common approaches, and the arrangement of the data forwarding paths considering security/privacy is another solution to clear this limitation, e.g., the forwarding paths are directly connected to those valid subscribers for private data.

## RELATED WORK

Related to the distributed stream data collection, various techniques have been proposed to disperse the communication loads for stream delivery (Shen et al., 2011; Win et al., 2018).

P2P stream delivery techniques have been proposed to use a P2P architecture and disperse the communication loads among the processing computers (nodes) (Zhang et al., 2005; Liao et al., 2006; Magharei et al., 2009; Yu et al., 2011; Sakashita et al., 2005). The P2P stream delivery techniques are divided into a pull type and a push type. In the pull type technique such as PPLive, DONet (Zhang et al., 2005), and SopCast, the reception nodes request data to other nodes and receive them. The reception nodes find the nodes which have not yet been received the requested data, and redundant communications do not occur. In the push type techniques such as AnySee, data are sent from the transmission nodes to other nodes (Liao et al., 2006). The transmission nodes find the nodes which have not yet received the requested data, and redundant communications do not occur. The techniques combining a pull type and a push type also have been proposed such as PRIME (Magharei et al., 2009).

Data delivery path construction techniques have been proposed as a multicast tree to prevent the concentration of communication loads to the specific node (Tran et al., 2003; Jin et al., 2007; Silawarawet et al., 2011; Le et al., 2012). In the ZIGZAG method, nodes construct clusters, and the multicast tree is constructed by the clusters (Tran et al., 2003). The number of clusters included in each depth of a multicast tree is made the same, and thus, the loads are dispersed. Multicast trees are constructed only of information gained in the application layer, and it is not necessary to understand the physical network structure.

In the MSMT/MBST method, the concentration of the communication loads is more prevented than the ZIGZAG method by consideration of the physical network structure (Jin et al., 2007). However, the MSMT/MBST method is not easy to be implemented because it is necessary to understand all the network structures between the nodes. In LAC (locality aware clustering), the loads are more dispersed than the ZIGZAG method by considering a part of nodes, though the physical network structure cannot be understood (Silawarawet et al., 2011).

In the above-described P2P stream delivery techniques, the same data streams are assumed to be sent to many reception nodes. In the delivery of the sensor data streams, however, the same sensor

data stream is assumed to have different delivery cycles to be delivered. In this case, those sensor data streams are delivered as different data streams for each delivery cycle. Thus, the communication loads cannot be efficiently dispersed. On the other hand, our proposed methods consider the different frequencies or cycles of each data stream and construct delivery paths to efficiently collect them.

As the distributed stream data collection systems, an existing method to reduce the number of messages to receive data from large-scale nodes is to execute a range query on key order-preserving overlays (Alaei et al., 2010; Gu et al., 2011; Legtchenko et al., 2012; Ohnishi et al. 2015; Shinomiya et al., 2011; Takeuchi et al., 2010). For example, “split forward broadcasting(SFB)” (Banno & Fujino et al., 2015) is an efficient way to construct tree structures for range queries. The data collection from publishers on a subscriber in TBPS corresponds to the execution of range query for nodes that have keys with a topic. It can reduce the number of messages by merging responses from nodes along the reverse path of the query delivery tree structure. However, this method loses the asynchronous real-time feature of TBPS. The latest sensor data is not delivered until the subscriber executes a range query. Once the tree structure is constructed, it can be reused, but the periodic execution of range queries is needed to catch up with the joins/leaves of publishers and subscribers.

Some existing works address the “aggregation problem” on structured overlays. DAT (Distributed Aggregation Tree) (Cai et al., 2007) constructs a tree to aggregate data from distributed nodes using the Chord (Stoica et al., 2003) overlay structure. DAT computes the aggregated value of all the local values applying a given aggregation function on the distributed nodes. DAT can be used for data collections if the nodes execute the message merging function as an aggregation function. However, to merge and collect published messages, the publishers need to publish messages at the same time, which is not a realistic assumption. Moreover, the aforementioned methods cause path concentration of data being forwarded on the nodes that are located close to the subscriber nodes on the overlay. In addition, once the tree structure is decided, it cannot be changed dynamically. As a result, a network process overload tends to occur on these nodes. On the other hand, our proposed ADCT method (Teranishi et al., 2017) can construct a flexible collection tree and adaptively adjust the maximum overhead for the nodes to merge and forward messages.

## CONCLUSION

In this paper, we proposed a skip graph-based collection scheme for sensor data streams considering phase differences. Our method uses phase shifting to avoid the load concentration to a specific time. Our simulation results show that the proposed scheme can equalize the number of the nodes even if the distribution of collection cycles is not uniform.

As future work, we will try to clear the current limitations described in the fifth section entitled Discussion. More specifically, we will consider other information to determine the data forwarding paths such as data types, node performances, and security/privacy.

## ACKNOWLEDGMENT

This research was supported by JSPS KAKENHI Grant Numbers 18K11316, G-7 Scholarship Foundation, and Research Grants from the University of Fukui.

## REFERENCES

- Alaei, S., Ghodsi, M., & Toossi, M. (2010). SkipTree: A new scalable distributed data structure on multidimensional data supporting range-queries. *Computer Communications*, 33(1), 73–82. doi:10.1016/j.comcom.2009.08.001
- Aspnès, J., & Shah, G. (2007). Skip graphs. *ACM Transactions on Algorithms*, 3(4), 1-25.
- Banno, R., Fujino, T., Takeuchi, S., & Takemoto, M. (2015). SFB: A scalable method for handling range queries on skip graphs. *IEICE Communications Express*, 4(1), 14–19. doi:10.1587/comex.4.14
- Banno, R., & Shudo, K. (2020). An efficient routing method for range queries in skip graph. *IEICE Transactions on Information and Systems*, 103D(3), 516-525.
- Banno, R., Takeuchi, S., Takemoto, M., Kawano, T., Kambayashi, T., & Matsuo, M. (2015). Designing overlay networks for handling exhaust data in a distributed topic-based pub/sub architecture. *Journal of Information Processing*, 23(2), 105–116. doi:10.2197/ipsjip.23.105
- Bessis, N., & Dobre, C. (2014). Big Data and Internet of Things: A roadmap for smart environments. *Studies in Computational Intelligence*, 546, 2014. doi:10.1007/978-3-319-05029-4
- Cai, M., & Hwang, K. (2007). Distributed aggregation algorithms with load-balancing for scalable grid resource monitoring. *Proceedings of the 21st IEEE International Parallel and Distributed Processing Symposium (IPDPS 2007)*, 1-10. doi:10.1109/IPDPS.2007.370313
- Castro, M., Druschel, P., Kermarrec, A.-M., & Rowstron, A. (2002). SCRIBE: A large-scale and decentralized application-level multicast infrastructure. *IEEE Journal on Selected Areas in Communications*, 20(8), 1489–1499. doi:10.1109/JSAC.2002.803069
- Eugster, P. T., Felber, P. A., Guerraoui, R., & Kermarrec, A.-M. (2003). The many faces of publish/subscribe. *ACM Computing Surveys*, 35(2), 114–131. doi:10.1145/857076.857078
- Gu, Y., & Boukerche, A. (2011). HD Tree: A novel data structure to support multi-dimensional range query for P2P networks. *Journal of Parallel and Distributed Computing*, 71(8), 1111–1124. doi:10.1016/j.jpdc.2011.04.003
- Hodges, S., Taylor, S., Villar, N., Scott, J., Bial, D., & Fischer, P. T. (2013). Prototyping connected devices for the Internet of Things. *IEEE Computer*, 46(2), 26–34. doi:10.1109/MC.2012.394
- Jin, X., Yiu, W.-P. K., Chan, S.-H. G., & Wang, Y. (2007). On maximizing tree bandwidth for topology-aware peer-to-peer streaming. *IEEE Transactions on Multimedia*, 9(8), 1580–1592. doi:10.1109/TMM.2007.907459
- Kawakami, T., Ishi, Y., Yoshihisa, T., & Teranishi, Y. (2017). A skip graph-based collection system for sensor data streams considering phase differences. *Proceedings of the 8th International Workshop on Streaming Media Delivery and Management Systems (SMDMS 2017)*, 506-513.
- Kawakami, T., Yoshihisa, T., & Teranishi, Y. (2018). A load distribution method for sensor data stream collection considering phase differences. *Proceedings of the 9th International Workshop on Streaming Media Delivery and Management Systems (SMDMS 2018)*, 357-367.
- Kawakami, T., Yoshihisa, T., & Teranishi, Y. (2019). Evaluation of a distributed sensor data stream collection method considering phase differences. *Proceedings of the 10th International Workshop on Streaming Media Delivery and Management Systems (SMDMS 2019)*, 444-453.
- Le, T. A., & Nguyen, H. (2012). Application-aware cost function and its performance evaluation over scalable video conferencing services on heterogeneous networks. *Proceedings of the IEEE Wireless Communications and Networking Conference: Mobile and Wireless Networks (WCNC 2012 Track 3 Mobile and Wireless)*, 2185-2190. doi:10.1109/WCNC.2012.6214155
- Legtchenko, S., Monnet, S., Sens, P., & Muller, G. (2012). RelaxDHT: A churn-resilient replication strategy for peer-to-peer distributed hash-tables. *ACM Transactions on Autonomous and Adaptive Systems*, 7(2).
- Liao, X., Jin, H., Liu, Y., Ni, L. M., & Deng, D. (2006). AnySee: Peer-to-peer live streaming. *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM 2006)*, 1-10.
- Magharei, N., & Rejaie, R. (2009). PRIME: Peer-to-peer receiver-driven mesh-based streaming. *IEEE/ACM Transactions on Networking*, 17(4), 1052–1065. doi:10.1109/TNET.2008.2007434

- Ohnishi, M., Inoue, M., & Harai, H. (2015). Incremental distributed construction method of Delaunay overlay network on detour overlay paths. *Journal of Information Processing*, 21(2), 216–224. doi:10.2197/ipsjip.21.216
- Rahimian, F., Girdzijauskas, S., Payberah, A. H., & Haridi, S. (2011). Vitis: A gossip-based hybrid overlay for Internet-scale publish/subscribe enabling rendezvous routing in unstructured overlay networks. *Proceedings of the 25th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2011)*, 746-757. doi:10.1109/IPDPS.2011.75
- Ratnasamy, S., Handley, M., Karp, R. M., & Shenker, S. (2001). Application-level multicast using content-addressable networks. *Proceedings of the 3rd International COST264 Workshop on Networked Group Communication (NGC 2001)*, 14-29. doi:10.1007/3-540-45546-9\_2
- Sakashita, S., Yoshihisa, T., Hara, T., & Nishio, S. (2010). 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)*, 166-172.
- Schütt, T., Schintke, F., & Reinefeld, A. (2008). Range queries on structured overlay networks. *Computer Communications*, 31(2), 280–291. doi:10.1016/j.comcom.2007.08.027
- Shao, X., Jibiki, M., Teranishi, Y., & Nishinaga, N. (2015). Hierarchy-aware skip graph for sensing resource discoveries on large-scale sensor overlay networks. *Computer Communications*, 70, 15–27. doi:10.1016/j.comcom.2015.06.021
- Shen, Z., Luo, J., Zimmermann, R., & Vasilakos, A. V. (2011). Peer-to-peer media streaming: Insights and new developments. *Proceedings of the IEEE*, 99(12), 2089–2109. doi:10.1109/JPROC.2011.2174089
- Shinomiya, J., Teranishi, Y., Harumoto, K., & Nishio, S. (2011). A sensor data collection method under a system constraint using hierarchical Delaunay overlay network. *Proceedings of the 7th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP 2011)*, 300-305. doi:10.1109/ISSNIP.2011.6146566
- Silawarawet, K., & Nupairoj, N. (2011). Locality-aware clustering application level multicast for live streaming services on the Internet. *Journal of Information Science and Engineering*, 27(1), 319–336.
- Stoica, I., Morris, R., Liben-Nowell, D., Karger, D. R., Kaashoek, M. F., Dabek, F., & Balakrishnan, H. (2003). Chord: A scalable peer-to-peer lookup protocol for Internet applications. *IEEE/ACM Transactions on Networking*, 11(1), 17–32. doi:10.1109/TNET.2002.808407
- Takeuchi, S., Shinomiya, J., Shiraki, T., Ishi, Y., Teranishi, Y., Yoshida, M., & Shimojo, S. (2010). A large scale key-value store based on range-key skip graph and its applications. *Proceedings of the 15th International Conference on Database Systems for Advanced Applications (DASFAA 2010)*, 432-435. doi:10.1007/978-3-642-12098-5\_43
- Teranishi, Y., Banno, R., & Akiyama, T. (2015). Scalable and locality-aware distributed topic-based pub/sub messaging for IoT. *Proceedings of the 2015 IEEE Global Communications Conference (GLOBECOM 2015)*. doi:10.1109/GLOCOM.2015.7417305
- Teranishi, Y., Kawakami, T., Ishi, Y., & Yoshihisa, T. (2017). A large-scale data collection scheme for distributed topic-based pub/sub. *Proceedings of the 2017 International Conference on Computing, Networking and Communications (ICNC 2017)*. doi:10.1109/ICNC.2017.7876131
- Tran, D. A., Hua, K. A., & Do, T. (2003). ZIGZAG: An efficient peer-to-peer scheme for media streaming. *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, 1283-1292. doi:10.1109/INFCOM.2003.1208964
- Win, E. K., Yoshihisa, T., Ishi, Y., Kawakami, T., Teranishi, Y., & Shimojo, S. (2018). A lost sensor data recovery scheme for sensor data stream multicasting. *Journal of Information Processing*, (26), 158-168.
- Yu, L., Liao, X., Jin, H., & Jiang, W. (2011). Integrated buffering schemes for P2P VoD services. *Peer-to-Peer Networking and Applications*, 4(1), 63–74. doi:10.1007/s12083-010-0094-y
- Zhang, X., Liu, J., Li, B., & Yum, T.-S. P. (2005). 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)*, 2102-2111.

*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. From April 2015 to February 2020, he was an assistant professor at Nara Institute of Science and Technology. Since March 2020, he has been a senior assistant professor at the University of Fukui. His research interests include distributed computing, rule-based systems, and stream data processing. He is a member of the Information Processing Society of Japan (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-on-demand, 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 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 National 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, IEICE, and IEEE.*