

A P2P Sensor Data Stream Delivery System to Accommodate Heterogeneous Cycles Using Skip Graphs

Tomoya Kawakami^{*†}, Yoshimasa Ishi[†], Tomoki Yoshihisa[†] and Yuuichi Teranishi^{†‡}

^{*}*Graduate School of Information Science, Nara Institute of Science and Technology*

Ikoma, Nara, Japan

kawakami@is.naist.jp

[†]*Cybermedia Center, Osaka University*

Ibaraki, Osaka, Japan

ishi.yoshimasa@ais.cmc.osaka-u.ac.jp, {yoshihisa, teranisi}@cmc.osaka-u.ac.jp

[‡]*National Institute of Information and Communications Technology*

Koganei, Tokyo, Japan

Abstract—In this paper, we propose a method using skip graphs to delivery sensor data streams with heterogeneous delivery cycles. Currently skip graphs have been proposed as one of structured overlay networks that construct links among nodes based on a specific rule. The proposed method sorts nodes by their delivery cycles and constructs delivery paths based on skip graphs. We confirmed in simulation that our proposed method can delivery sensor data with heterogeneous cycles using skip graphs to distribute the load of source node.

I. INTRODUCTION

The Internet of Things (IoT) means that objects such as computing devices, electrical appliances and sensors connect to the Internet and interact each other to collaborate and realize various intelligent services. Especially sensors have an important role in the IoT, and the sensors generate observed data periodically. The continuous periodical data generated by the sensors is called ‘sensor data stream.’ In the IoT services, a huge amount of sensor data streams are required to delivered to appropriate destinations such as users and processes. As for this sensor data stream delivery, the destinations are possible to require different delivery cycles for the same sensor data stream based on various reasons such as a performance of the receiver, network environments and applications. In the case where a live video of a solar eclipse taken from a camera is delivered, for example, the video is delivered at 30 fps to personal computer users connected to the Internet through a wire and is delivered at 10 fps to personal computer users connected to the Internet through a 3G channel while moving.

It is general 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 [1]–[9]. In these researches, in the case where the same sensor data stream is delivered to a number of terminals (destinations), the communication load

of the source is dispersed by what the destinations send the received data to other destinations. When the delivery cycle is different, the sensor data stream whose delivery cycle is a common divisor of required cycles can be delivered to all of the destinations if the delivery cycles are in a multiple relationship or can be approximated as having a multiple relationship. However, the destinations receive redundant data which are not included to the times of each required cycle.

We have proposed P2P-based methods to construct scalable and efficient sensor data stream system that accommodates different delivery cycles by distributing communication loads of the nodes [10]. In the existing methods, destinations having a long delivery cycle relay the sensor data stream to other destinations so that the load of the source is dispersed.

Currently skip graphs have been proposed as one of structured overlay networks that construct links among nodes based on a specific rule [11]. In this paper, we propose a method using skip graphs to delivery sensor data streams with heterogeneous delivery cycles. The proposed method sorts nodes by their delivery cycles and constructs delivery paths based on skip graphs. We discuss two approaches that sort in ascending order and descending order.

II. ADDRESSED PROBLEMS

A. Assumed Environment

The purpose of the present study is to disperse the communication load in the sensor stream deliveries having different delivery cycles. The source nodes have sensors so as to gain sensor data periodically. The destination node which wants to receive sensor data searches for the corresponding source node and requires a sensor data stream having a delivery cycle that is desired to be received. Upon reception of the query from the destination node, the source node determines the delivery path from the sensor data stream being delivered. The queries are received during delivery, and whenever a query is received, the delivery path

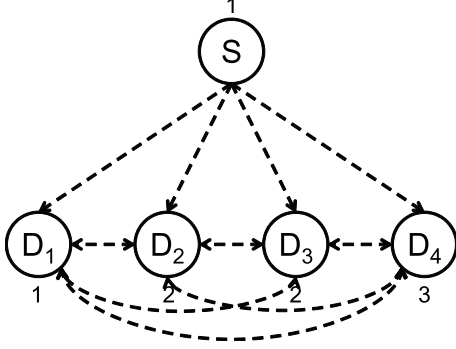


Figure 1: An example of input setting

Table I: An example of the sensor data delivery

Time	1	2	3	4	5	6	7	...
D_1 (Cycle=1)	o	o	o	o	o	o	o	...
D_2 (Cycle=2)		o		o		o		...
D_3 (Cycle=2)		o		o		o		...
D_4 (Cycle=3)			o			o		...

is changed. The sensor data stream is delivered along the determined path, and when a destination node sends a sensor data stream to another destination node, a query is received by the destination node to which the sensor data stream is to be sent. The delivery path changes whenever the source node receives a query for delivering a sensor data stream.

B. Input Setting

The source node of sensor data is S and N destination nodes are D_i ($i = 1, \dots, N$). In addition, the delivery cycle of S is C_0 and the delivery cycle required by D_i is C_i . The sensor data that has not been gained by the source node cannot be delivered, and therefore, C_i is a multiple of C_0 , which can be represented by $C_i = n_j C_0$ using a certain integer n_j ($= 1, 2, \dots$).

In Figure 1, each node indicates a source node and the branches indicate delivery 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 deliver a sensor data stream depending on the delivery method. The source node S is at the top and the four destination nodes D_1, \dots, D_4 ($N = 4$) are at the bottom. The figure in the vicinity of each destination node indicates the delivery cycle, and $C_0 = 1$, $C_1 = 1$, $C_2 = 2$, $C_3 = 2$ and $C_4 = 3$. This corresponds to the case where a live camera acquires an image once every second, and D_1 views the image once every second, D_2 and D_3 view the image once every two seconds, and D_4 views the image once every three seconds, for example. Table I shows the delivery cycle of each destination node and the sensor data to be received in the example in Figure 1.

C. Objective Function

The communication load of S is L_0 and the communication load of D_i is L_i . The communication load SL of the entirety of the system is given by the following equation:

$$SL = \sum_{i=0}^N L_i \quad (1)$$

In addition, the following fairness index (FI) is often used as an index for load dispersion:

$$FI = \frac{\left(\sum_{i=0}^N L_i\right)^2}{N \sum_{i=0}^N L_i^2} \quad (2)$$

where $0 \leq FI \leq 1$, and when $FI = 1$, $L_0 = \dots = L_N$. It is indicated that the closer FI is to 1, the more the load is dispersed.

Another purpose of the present study is to disperse the communication load to the destination nodes while suppressing the communication load of the entirety of the system. Therefore, the objective function is SL and $1 - FI$, and the delivery path is determined to make these values minimum. In the present problem, the received sensor data stream can be sent to another destination node, and each destination node determines the sensor data to be sent.

D. Definition of Load

The communication load of the source node and the destination nodes 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 D_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 D_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 D_p to D_q ($q \neq p$; $p, q = 1, \dots, N$) is $R(p, q)$, and the number delivered by S to D_q is $R(0, q)$.

In the present study, the load with which one piece of sensor data can be received and sent per unit hour is normalized to 1, and the communication load L_r of D_r is given in the following equations:

$$L_r = I_r + O_r \quad (3)$$

$$I_r = \alpha \sum_{i=0}^N R(i, r) \quad (4)$$

$$O_r = \beta \sum_{i=0}^N R(r, i) \quad (5)$$

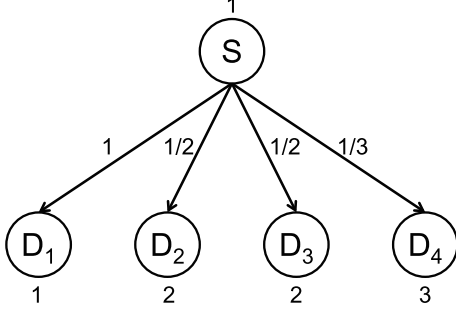


Figure 2: A case of delivering the sensor data stream directly from the source node

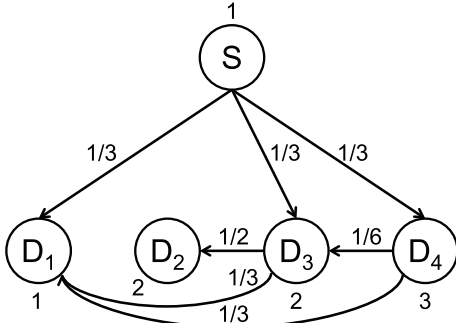


Figure 3: An example of LCF method

where α and β are loads with which one piece of sensor data is received and sent, respectively.

Figure 2 shows a case where $\alpha = \beta$ and the sensor data is delivered directly from the source node in the example in Figure 1. We call this method SD (Server Direct). The figures in the vicinity of the branches are the number of sensor data pieces per unit hour of the sensor data stream. In this example, the sensor data stream is delivered only from the server, and therefore, $R(0, q) = 1/C_q$ and $R(p, q) = 0$. Thus, $R(0, 1) = 1$, $R(0, 2) = 1/2$, $R(0, 3) = 1/2$, and $R(0, 4) = 1/3$. The load at each end is $L_0 = R(1, 0) + R(2, 0) + R(3, 0) + R(4, 0) + R(0, 1) + R(0, 2) + R(0, 3) + R(0, 4) = R(0, 1) + R(0, 2) + R(0, 3) + R(0, 4) = 2.33$, $L_1 = 1.00$, $L_2 = 0.500$, $L_3 = 0.500$, and $L_4 = 0.333$. In this case, $SL = 4.667$ and $FI = 0.617$. On the other hand, we have proposed P2P-based methods to construct scalable and efficient sensor data stream system that accommodates different delivery cycles by distributing communication loads of the nodes [10]. In the existing methods, destinations having a long delivery cycle relay the sensor data stream to other destinations so that the load of the source is dispersed. Figure 3 shows an example of the existing method called LCF (Longest Cycle First). In this case, $SL = 4.667$ and $FI = 0.992$, and thus, the load is dispersed without changing the load of the entirety of the system as compared to the case of Fig. 2.

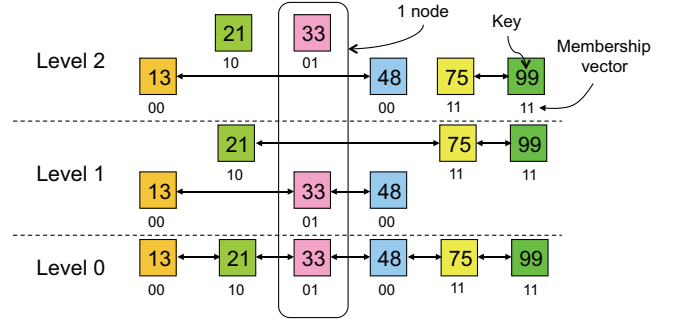


Figure 4: Skip graphs

III. PROPOSED METHOD USING SKIP GRAPHS

In this paper, we propose a method using skip graphs to delivery sensor data streams with heterogeneous delivery cycles.

A. Idea

Skip graphs have been proposed as one of structured overlay networks that construct links among nodes based on a specific rule [11]. Skip graphs are an application of skip lists to a P2P model. Figure 4 shows an example of skip graphs. In Fig. 4, squares show entries on routing tables of peers (nodes). The numbers inside the squares show keys of peers, and peers are ordered by the key values. The numbers below entries are called “membership vector.” Each peer has a membership vector and construct hierarchical lists with other peers based on a membership vector. The average number of hops to search peers based on a key is $\log n$ in skip graphs while n denotes the number of peers. Currently many researchs exist to enhance the skip graphs [12]–[14].

In this paper, we propose a method using hierarchical links of skip graphs to delivery sensor data streams with heterogeneous delivery cycles. The proposed method sorts nodes by their delivery cycles as keys and constructs hierarchical links based on skip graphs. The delivered nodes at each time are searched by cycles as keys. In this paper, we discuss two approaches that sort keys in ascending order and descending order.

B. Construction of delivery paths

1) *Delivery from the shorter nodes:* Figure 5 shows delivery paths in the case where one of the shortest cycle nodes receives sensor data from the source node first and each node sends the sensor data to the same or longer cycle nodes. In Fig. 5, selectable delivery cycles are 1, 2 and 3. Delivery cycle 1 and 3 have four nodes individually, and delivery cycle 2 has eight nodes. Since the least common multiple of the delivery cycles is six, the delivery paths are from time 1 to 6. Circles in Fig 5 show nodes. In this approach, each node constructs a hierarchical routing table based on skip graphs in ascending order. The key is

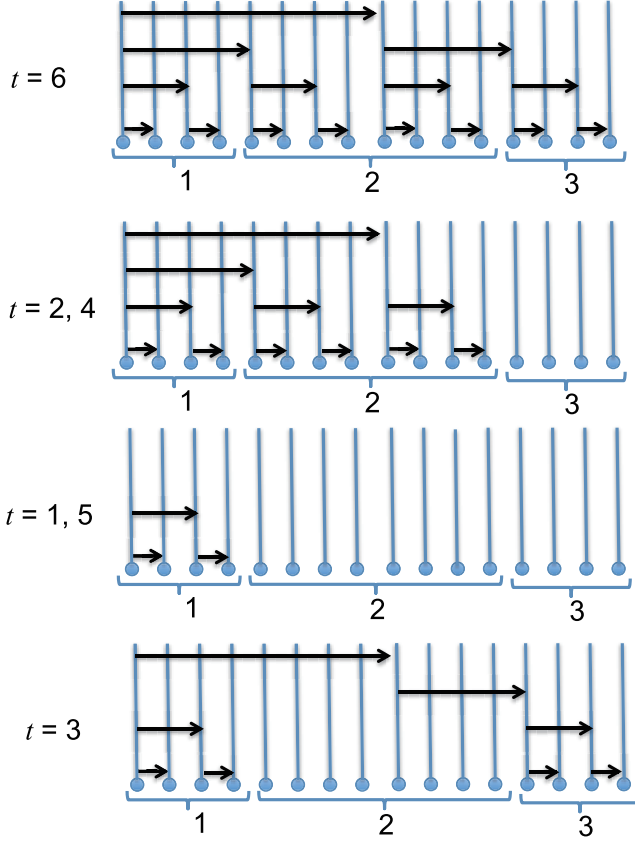


Figure 5: An example of delivery from the shorter cycle nodes

the selected delivery cycle. We call this approach SCF-SG (Shortest Cycle First on Skip Graphs).

In Fig. 5, one of the nodes in cycle 1 receives sensor data from the source node first at each time. After that, the sensor data are sent to the same or longer cycle nodes recursively based on skip graphs. Sending or receiving nodes are different at each time, and all nodes receive sensor data at time 6.

2) *Delivery from the longer nodes:* Figure 6 shows delivery paths in the case where one of the longest cycle nodes receives sensor data from the source node first and each node sends the sensor data to the same or shorter cycle nodes. In Fig. 6, selectable delivery cycles and the number of nodes are same to Fig. 5. In this approach, each node constructs a hierarchical routing table based on skip graphs in descending order. We call this approach LCF-SG (Longest Cycle First on Skip Graphs).

In Fig. 6, one of the nodes in cycle 3 receives sensor data from the source node first at each time. After that, the sensor data are sent to the same or shorter cycle nodes recursively based on skip graphs.

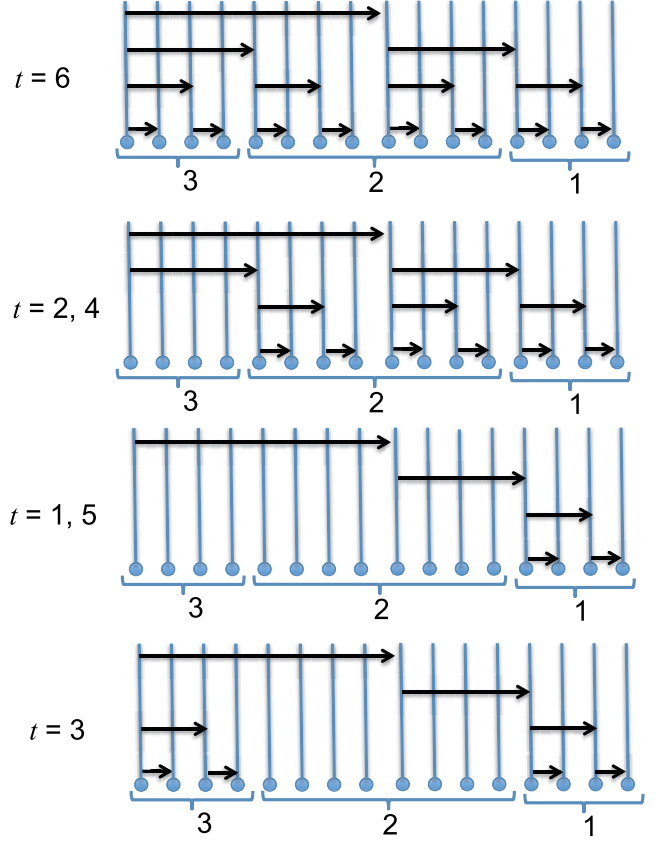


Figure 6: An example of delivery from the longer cycle nodes

IV. EVALUATION

We evaluate the proposed method using Skip Graph by simulation.

A. Simulation Environment

The simulation environment is the same as the environment in Fig. 6. The number of source nodes is 1, and the number of destination nodes is 16. About the delivery cycles selected by each destination node, four nodes select cycle 1, another eight nodes select cycle 2 and the rest four nodes select cycle 3. The simulation time is from one to six, which is the least common multiple of the selected delivery cycles.

As comparison methods for the proposed SCF-SG method and LCF-SG method, we use SD method, SCF method and LCF method described in Section II.

B. Simulation Results

Figure 7 shows the total loads of a source node and destination nodes. In SD method, SCF method and LCF method, sensor data are sent only to the related nodes on each time. Therefore, the total loads in SD method, SCF method and LCF method show the value in the lowest case. On the other hand, in SCF-SG method and LCF-SG

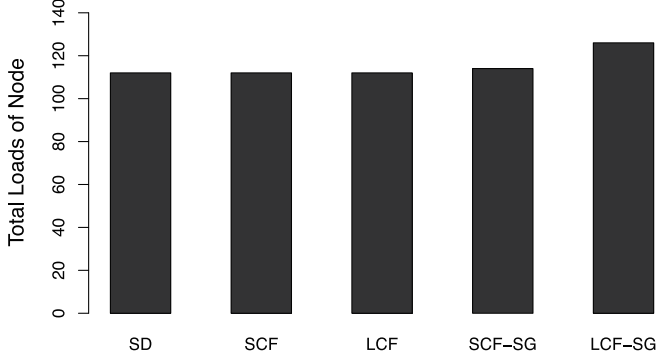


Figure 7: System total loads

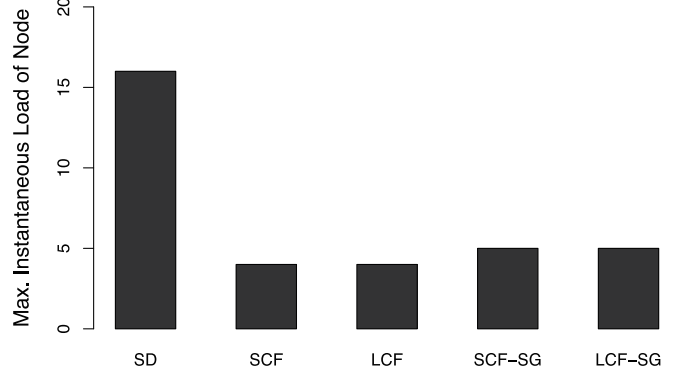


Figure 9: Maximum instantaneous load of node

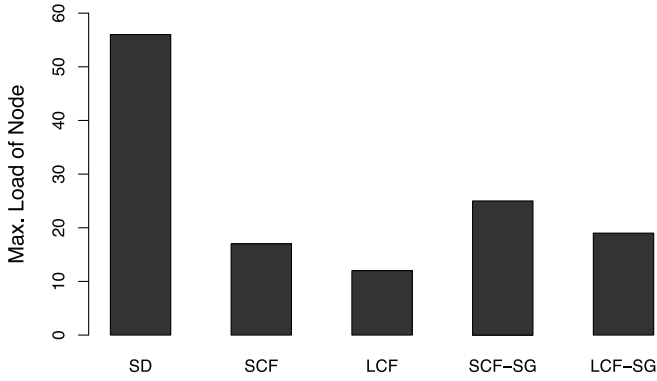


Figure 8: Maximum load of node

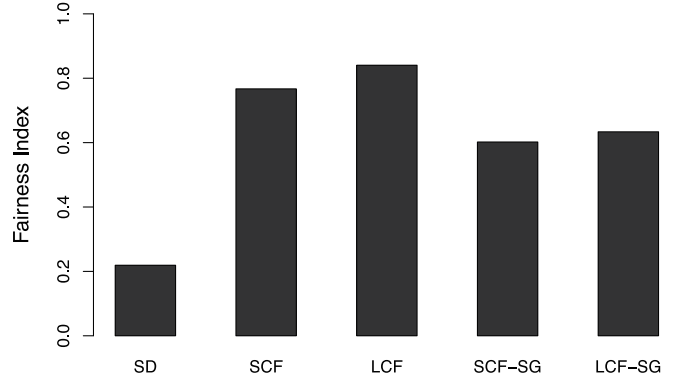


Figure 10: Fairness Index

method, nodes are probable to relay any sensor data to other nodes via skip graphs. Therefore, the total loads in SCF-SG method and LCF-SG method are higher than the value on SD method, SCF method or LCF method. However, this increase of the total loads seems to be solved by changing the priority of the relay nodes based on those relations to each time.

Figure 8 shows the maximum load of node from time 1 to 6. In SD method, the load of the source node is the maximum value since the source node sends sensor data to all destination nodes. On the other hand, SCF method, LCF method, SCF-SG method and LCF-SG method can reduce the maximum load of node since destination nodes relay sensor data from the source node to other destination nodes. In addition, in SCF-SG method and LCF-SG method, some nodes send many sensor data such as the leftmost node in Fig. 5 or 6. However, this imbalance seems to be reduced by determining the order of nodes among same cycle at random.

Figure 9 shows the maximum instantaneous load of node from time 1 to 6. The maximum instantaneous load is the maximum load for each node and time. Similar to Fig. 8, in SD method, the load of the source node at time 6 is the maximum value since all cycles relate to time 6 and the source node sends sensor data to all destination nodes at time 6. Also SCF method, LCF method, SCF-SG method

can LCF-SG method can reduce the maximum instantaneous load like Fig. 8.

Figure 10 shows *FI* among loads of each node from time 1 to 6. *FI* in SD method is the lowest since the load of the source node is extremely high, described above. On the other hand, *FI* in SCF method, LCF method, SCF-SG and LCF-SG method are high since the load of the source node is distributed to other nodes. Similar to Fig. 8, *FI* in SCF-SG method and LCF-method are lower than *FI* in SCF method and LCF method. However, *FI* in SCF-SG method and LCF-method seems to be increased by determining the order of nodes among same cycle at random.

Figure 11 shows the maximum number of hops of sensor data from time 1 to 6. The number of hops in SD method is always one since the source node sends sensor data to all destination nodes. The maximum numbers of hops in SCF method and LCF method depend on the number of nodes that has same cycles since sensor data are relayed among same cycle nodes. The maximum numbers of hops in SCF-SG method and LCF-SG method depend on the number of nodes, and the average number of hops to a specific node is $\log n$ by skip graphs. Therefore, the maximum numbers of hops in SCF-SG method and LCF-SG method are lower than the maximum number hops in SCF method and LCF method.

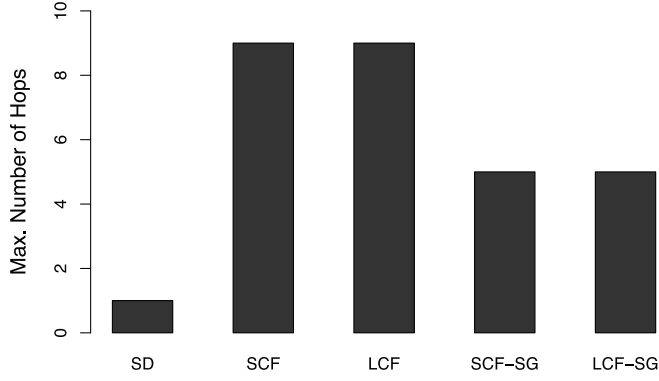


Figure 11: Maximum number of hops

V. CONCLUSION

In this paper, we proposed a method using skip graphs to delivery sensor data streams with heterogeneous delivery cycles. The proposed method sorts nodes by their delivery cycles and constructs delivery paths based on skip graphs. We discussed two approaches that sort in ascending order and descending order. In addition, we confirmed in simulation that our proposed method can delivery sensor data with heterogeneous cycles using skip graphs to distribute the load of source node.

In the future, we plan to evaluate the proposed method in various environments.

ACKNOWLEDGMENT

This research was partly supported by the collaborative research of National Institute of Information and Communications Technology (NICT) and Osaka University (Research on high functional network platform technology for large-scale distributed computing). This research was partly supported by the Strategic Information and Communications R&D Promotion Programme (SCOPE) of the Ministry of Internal Affairs and Communications.

REFERENCES

- [1] X. Zhang, J. Liu, B. Li, and T.-S. P. Yum, "CoolStreaming/DONet: A data-driven overlay network for peer-to-peer live media streaming," in *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005)*, Mar. 2005, pp. 2102–2111.
- [2] X. Liao, H. Jin, Y. Liu, L. M. Ni, and D. Deng, "Anysee: Peer-to-peer live streaming," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM 2006)*, Apr. 2006, pp. 1–10.
- [3] N. Magharei and R. Rejaie, "PRIME: Peer-to-peer receiver-driven mesh-based streaming," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM 2007)*, May 2007, pp. 1415–1423.
- [4] 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.
- [5] S. Sakashita, T. Yoshihisa, T. Hara, and S. Nishio, "A data reception method to reduce interruption time in P2P streaming environments," in *Proceedings of the 13th International Conference on Network-Based Information Systems (NBIS 2010)*, Sep. 2010, pp. 166–172.
- [6] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2002)*, Aug. 2002, pp. 205–217.
- [7] D. A. Tran, K. A. Hua, and T. Do, "ZIGZAG: An efficient peer-to-peer scheme for media streaming," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, vol. 2, Mar. 2003, pp. 1283–1292.
- [8] 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 Transactions on Multimedia*, vol. 9, no. 8, pp. 1580–1592, Dec. 2007.
- [9] 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.
- [10] 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, Jul. 2014.
- [11] J. Aspnes and G. Shah, "Skip graphs," *ACM Transactions on Algorithms (TALG)*, vol. 3, no. 4 (37), Nov. 2007.
- [12] Y. Shu, B. C. Ooi, K.-L. Tan, and A. Zhou, "Supporting multi-dimensional range queries in peer-to-peer systems," in *Proceedings of the 5th IEEE International Conference on Peer-to-Peer Computing (P2P 2005)*, Aug. 2005, pp. 173–180.
- [13] M. T. Goodrich, M. J. Nelson, and J. Z. Sun, "The rainbow skip graph: A fault-tolerant constant-degree distributed data structure," in *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithm (SODA 2006)*, Jan. 2006, pp. 384–393.
- [14] R. Banno, T. Fujino, S. Takeuchi, and M. Takemoto, "SFB: A scalable method for handling range queries on skip graphs," *IEICE Communications Express*, vol. 4 (2015), no. 1, pp. 14–19, Feb. 2015.