Regular Paper

A P2P-Based Sensor Data Stream Delivery Method to Accommodate Heterogeneous Cycles

Tomoya Kawakami[1,a)]   Yoshimasa Ishi[2]   Tomoki Yoshihisa[2]   Yuuichi Teranishi[2,3]

Abstract: Due to the prevalence of sensors such as security cameras or environmental monitoring, sensor data stream delivery which means delivering the observed data continuously attracts great attention. For sensor data stream delivery, various methods to distribute communication loads in the case of delivering the same sensor data streams to multiple clients have been studied. Although these methods assume that the sensor data streams have the same data delivery cycles, data delivery cycles sometimes differ. Hence, we propose a P2P-based method to distribute communication loads in the case of delivering the sensor data streams that have different data delivery cycles. The proposed method distributes communication loads by redelivering the sensor data that have the same delivery time but are included in different sensor data streams. We evaluated the effectiveness of the proposal in simulations and confirmed the load distribution in the case of many receivers at the same time.

Keywords: sensor data, data stream, delivery cycle, distributed processing, load estimation

1. Introduction

In recent years, various types of applications such as video delivery and environmental monitoring have been possible, and therefore, sensor data stream delivery where sensor data are periodically gained and delivered continuously has been attracting great attention. As for this sensor data stream delivery, it is possible for the same sensor data stream to have different delivery cycles depending on the receivers.

For example, the following deliveries are possible;

- A live video of a solar eclipse is delivered at 30 fps to personal computer users connected to the Internet through a wire. On the other hand, the video is delivered at 10 fps to mobile computer users connected to the Internet through a 3G channel.
- When some computers forecast the future temperature from the existing temperature data, the delivery cycle to each computer is determined based on the processing speed.
- When the user is continuously checking the amount of rain to decide the timing to go out in the rainy day, the data are delivered once per second to a smart phone if it connects to a power source. On the other hand, the data are delivered once per minute in order to reduce power consumption if it does not connect to a power source.

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 node) have been studied in the data streaming [1], [2], [3], [4], [5], [6], [7], [8], [9], [10]. In these researches, in the case where the same sensor data stream is delivered to a number of terminals (destination nodes), the communication load of the source node is dispersed when the destination nodes send the received data to other destination nodes. When the delivery cycle is different, the sensor data stream whose delivery cycle is a common divisor of the required cycles can be delivered to all of the destination nodes if the delivery cycles are in a multiple relationship or can be approximated as having a multiple relationship. However, the destination nodes receive redundant data which are not included to the times of each required cycle.

Therefore, the present study proposes a delivery technique according to which the communication load is dispersed by taking the delivery cycle of each destination node into consideration when sensor data streams having different delivery cycles are delivered. The main contributions of this paper are as follows:

- Tackles to a problem of redundant data delivery for the P2P-based sensor data delivery on clouds where heterogeneous cycles of requests co-exist, that has never been treated in the previous studies.
- Proposes a P2P-based delivery algorithm LLF-H, which can both keep the maximum communication load low and distributed. It also guarantees the number of hops in the P2P delivery, thus can keep delivery latency low.

In the following, the problems addressed by the present study are defined in Section 2. The proposed technique is described in Section 3, its evaluation is presented in Section 4. We describe the related work in Section 5. Finally, the conclusion of the study is presented in Section 6.

1   Graduate School of Engineering, Kobe University, Kobe, Hyogo 657–8501, Japan
2   Cybermedia Center, Osaka University, Ibaraki, Osaka 567–0047, Japan
3   National Institute of Information and Communications Technology, Koganei, Tokyo 184–8795, Japan
a)   kawakami.tomoya@eedept.kobe-u.ac.jp

2.   Addressed Problems

2.1   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 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.

2.2   Input Setting

The source node of sensor data is S and N destination nodes are D_i ($i = 1, \cdots, N$). 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 ($= 1, 2, \cdots$).

In **Fig. 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 $D_1, \cdots, D_4$ ($N = 4$) 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 1** shows the delivery cycle of each destination node and the sensor data to be received in the example in Fig. 1.
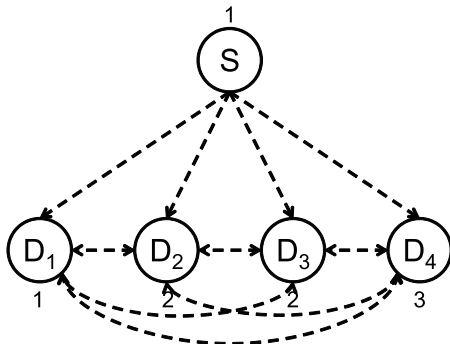

Fig. 1   An example of input setting.

2.3   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:

$$SL = \sum_{i=0}^{N} L_i \tag{1}$$

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

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

where $0 \leq FI \leq 1$, and when $FI = 1$, $L_0 = \cdots = L_N$. 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.

2.4   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 ($q \neq p$; $p, q = 1, \cdots, N$) 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:

$$L_r = I_r + O_r \tag{3}$$

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

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

Table 1   An example of the sensor data delivery.

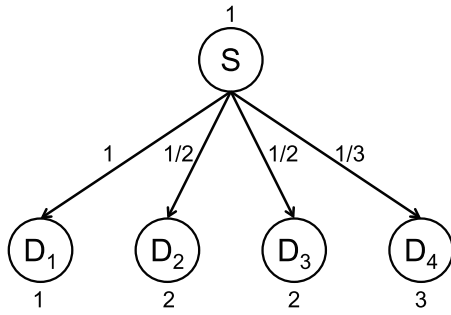| Time	1	2	3	4	5	6	7	...
D_1 (Cycle=1)	○	○	○	○	○	○	○	...
D_2 (Cycle=2)		○		○		○		...
D_3 (Cycle=2)		○		○		○		...
D_4 (Cycle=3)			○			○		...

Fig. 2   A case of delivering the sensor data stream directly from the source node.

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 Fig. 1. 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 $R(0, 3) = 1/3$. 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 $FI = 0.617$.

3.   Proposed P2P Delivery Method

3.1   Basic Idea

In the present research, the sensor data delivered at the same time and included in different sensor data streams are redelivered, and thus, the communication load is dispersed. In the case of Fig. 1 and Table 1, the delivery cycle at D_1 is 1, and all of the sensor data gained by the sensor are received. The delivery cycle at D_2 and D_3 is 2, and the sensor data are received every cycle of time 2. Likewise, D_4 receives sensor data every cycle of time 3, and the sensor data at time 6 is included in all the sensor data streams having the cycles 1, 2, and 3. As a result, the destination node having a cycle 2 delivers sensor data at time 6 to the destination nodes having a cycle 1 or 3, or the destination node having a cycle 3 delivers sensor data at time 6 to the destination node having a cycle 1 or 2 so that the sensor data can be redelivered without direct delivery from the server. In the case where the server delivers directly, the communication load is concentrated on the server. As in this example, however, the node that has received the sensor data can redeliver it to another node so that the communication load can be dispersed.

In our assuming environment, the reception load of a long cycle node is lower than the load of a short cycle node. As a simple method for dispersing the load through redelivery, the method that the longest cycle node in each time receives from source node and redeliver to other destination nodes is possible. However, the transmission loads of the long cycle nodes become higher when there are many other destination nodes at the same time, and the loads between the nodes delivered with a medium cycle are unbalanced. In addition, the delivery path is determined on the basis of the number of transmissions and receptions of data, and there-

fore, the effects of the load distribution are low in an environment where the load cannot be simply represented by the number of transmissions and receptions.

3.2   Approach

In the present research, we propose a P2P-based method which estimates loads of each nodes and the lowest load node at points in each time sends the data to another destination node. In the proposed method, the communication load is dispersed when the destination nodes of which the assumed load is small sends the communication load to the sensor data of the same delivery time that is included in different sensor data streams. In addition, a value that can be calculated from any model is defined as a load, for example, α and β are given to the number of transmissions and receptions of the sensor data as coefficients, in order to determine the delivery path for the purpose of dispersing various values. Moreover, the number of hops until each destination node receives data is important because the number of hops affects the delay of data delivery. In the proposed method, the maximum number of hops allowable to receive data is specified, and delivery paths satisfying the limitation of hops are determined. We call this proposed method LLF-H (lowest load first considering hops).

The load is estimated and the delivery path is determined before the start of delivery, and the following contents are delivered to each destination node after the delivery path has been determined. The contents are about the sensor data sent and received by each destination node at each point in time, and each destination node sends the received sensor data to another destination node according to the timetable. In the timetable, the time is set as the least common multiple of the delivery cycles of all the destination nodes after time 1. After that, the time returns to time 1 and transmissions following the timetable are repeated.

In order to generate the timetable, the amount of calculation becomes enormous due to a large number of combinations, and therefore, in the LLF-H method, a restriction is set so that the sensor data stream is sent from a node having a longer cycle to a node having a shorter cycle. As a result of this restriction, the source node first sends the sensor data stream to the destination node having the longest cycle at each time. Likewise, the destination node having the longest cycle at each time sends the sensor data stream to the destination node having the second longest cycle. From the above description, the following loads can be assumed on the basis of the delivery cycles of the destination nodes so as to be used for the preparation of the timetable of the delivery path.

- Reception load within the least common multiple cycle of each destination node
- Transmission load from the source node to the destination node having the longest cycle at each time
- Transmission load from the destination node having the longest cycle to the destination node having the second longest cycle

3.3   Load Estimation and Delivery Path Determination

On the basis of the approach in Section 3.2, the delivery procedure in the LLF-H method is described below. **Figure 3** shows

Require:
    cycles: Arrangement of Delivery Cycles of the Nodes Sorted in Ascending Order (cycle of source node is −1 and at index 0)
   *allowableHopCount*: Allowable Maximum Number of Hops

```
 1: loads; // Arrangement of loads of each node in cycles
 2: hopCounts; // Arrangement of the number of hops until each node in
    cycles receives data
 3: cycLcm ← calculateLCM(cycles); // The least common multiple of the
    delivery cycles of the destination nodes is calculated
 4: for i ← 1 to cycles.length do
 5:    loads[i] ← α * cycLcm/cycles[i]; // The reception load of each desti-
       nation node is calculated
 6: end for
 7: for i ← 1 to cycLcm do
 8:    longNode1st ← getLongestNodeIndex(cycles, i); // The index of desti-
       nation node having the longest cycle at time i is picked
 9:    if longNode1st ≠ 0 then
10:       requestToSend(0, longNode1st, i); // Delivery path from the source
          node to the destination node having the longest cycle node
11:       loads[0] ← loads[0] + β; // The transmission load is added to the
          source node
12:    end if
13:    longNode2nd ← get2ndLongestNodeIndex(cycles, i); // The index of
       destination node having the second longest cycle at time i is picked
14:    if longNode2nd ≠ longNode1st then
15:       requestToSend(longNode1st, longNode2nd, i);  // Delivery path
          from the destination node having the longest cycle to the destina-
          tion node having the second longest cycle
16:       loads[longNode1st] ← loads[longNode1st] + β; // The transmis-
          sion load is added to the destination node having the longest cycle
17:    end if
18: end for
19: for i ← 1 to cycLcm do
20:    hopCounts[0] ← 0; // Set 0 to the number of hops of the source node
21:    longNode1st ← getLongestNodeIndex(cycles, i);
22:    hopCounts[longNode1st] ← hopCounts[0] + 1; // Set 1 to the number
       of hops of the longest cycle node
23:    longNode2nd ← get2ndLongestNodeIndex(cycles, i);
24:    hopCounts[longNode2nd] ← hopCounts[longNode1st] + 1; // Set 2 to
       the number of hops of the second longest cycle node
25:    for j ← longNode2nd − 1 to 1 do
26:       if i mod cycles[j] = 0 then
27:          minNode ← longNode1st;
28:          minLoad ← loads[minNode];
29:          for k ← longNode2nd to j + 1 do
30:             tmpLoad ← loads[k];
31:             if i mod cycles[k] = 0
                and hopCounts[k] < allowableHopCount
                and tmpLoad < minLoad then
32:                minLoad ← tmpLoad;
33:                minNode ← k;
34:             end if
35:          end for
36:          requestToSend(minNode, j, i); // Delivery path from the desti-
             nation node having the minimum load to Dj at time i
37:          loads[minNode] ← loads[minNode] + β; // The transmission
             load is added to the destination node having the minimum load
38:          hopCounts[j] ← hopCounts[minNode] + 1; // Set the number
             of hops from the lowest load node
39:       end if
40:    end for
41: end for
```

Fig. 3   Pseudocode for load estimation and delivery path determination.

the algorithm of the load estimation and the determination of the delivery paths. In the present paper, the example in Fig. 1 and Table 1 is used. The transmission and reception load per sensor data piece is $\alpha = \beta = 1$ and the maximum number of hops allowable in LLF-H is 3.

First, the delivery cycles of the destination nodes D_1, D_2, D_3 and D_4 are $C_1 = 1$, $C_2 = 2$, $C_3 = 2$ and $C_4 = 3$, respectively, and therefore, the least common multiple is 6. In this case, according to the LLF-H method, the delivery path of the sensor data is determined at each time from time 1 to time 6. Therefore, each
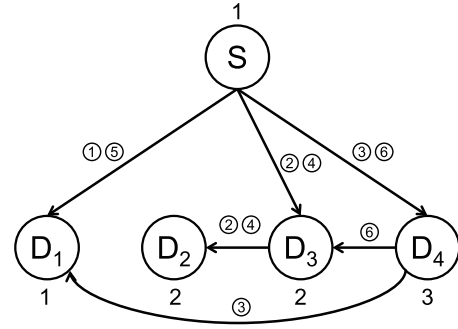
Fig. 4   Delivery paths determined by delivery cycles.

destination node finds the product of the number of received sensor data pieces and the reception load α per sensor data piece as the reception load from time 1 to time 6. This process is described from line 4 to line 6 of Fig. 3. The number of received sensor data pieces on each destination node is calculated by dividing the least common multiple by the delivery. For example, the number of reception data pieces of D_2 with a cycle of 2 is $6/2 = 3$. After calculating the reception load of each destination node, the transmission load from the source node to the destination node having the longest cycle is added. This process is described from line 8 to line 12 of Fig. 3. In addition, the transmission load from the destination node having the longest cycle to the destination node having the second longest cycle at each time is added. This process is described from line 13 to line 17 of Fig. 3. described in the line 16 of Fig. 3. The transmission load is the product of the number of sent sensor data pieces and the transmission load β per sensor data piece. **Figure 4** shows the delivery path that has been determined at this stage, until the line 18 of Fig. 3. The figures in the small circles in the vicinity of the delivery path in Fig. 4 indicate the sensor data at that time.

Next, the destination node that receives the sensor data at each time is picked, and a selection is made from combinations of a number of delivery paths so that the destination node having the smallest load sends the sensor data. According to the LLF-H method, the load estimation and the delivery path determination at the respective times are carried out in ascending order from the earlier time, described at line 19 of Fig. 3. In addition, the destination nodes at the respective times are picked in descending order from the destination node having the longer cycle, described at line 25 of Fig. 3. In the flow after Fig. 4, first, the delivery path to D_1 at time 1 has already been determined. Therefore, the procedure moves to the case at time 2, and thus, the delivery path to D_1 is first determined because the delivery path to D_3, D_2 have already been determined. **Figure 5** shows an example of the selection of a delivery path to D_2 in the case at time 6. In the example in Fig. 5, D_3 and D_4 can send the sensor data to D_2 at time 6 as indicated by the broken arrows. The loads of D_3 and D_4 at this point in time are $L_3 = 5/6$ and $L_4 = 4/6$. Therefore, the delivery path from D_4 having the smaller load is selected. The initial state to search the minimum load node is the longest cycle node because the longer cycle node tends to have a low load. This process is described from line 27 to line 35 of Fig. 3. After selecting the destination node to send, the delivery path is added to the timetable, described at line 36 of Fig. 3. Even if the deliv-
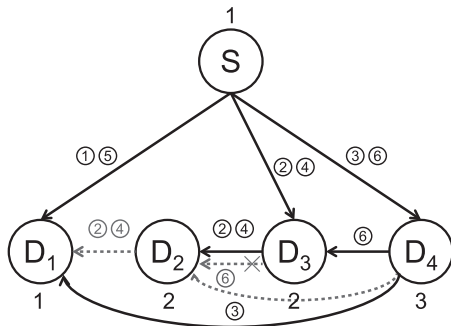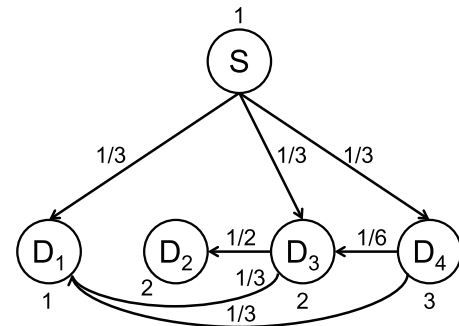
Fig. 5   Delivery path determination for D_2 at 6.

Fig. 6   Timetable for sensor data delivery.

ery path to the sender node has not been added at this moment, the path from the sender node is added because all delivery paths are finally added. This determination is reflected in the load of D_4, and the procedure moves to the selection of the next destination node or the delivery path at that time. Finally, the path to the destination node having the shortest cycle at the time of the least common multiple is determined, and then the procedure is complete. In the example in Fig. 4, at the point in time when the delivery path to D_1 is determined at time 6, the procedure is complete. **Figure 6** shows the final timetable in the example in Fig. 5. The solid arrows in Fig. 6 are paths that have been determined by the delivery cycle of each destination node, and the broken arrows are paths selected through the load estimation. In Fig. 6, $SL = 4.667$, $FI = 0.992$.

4.   Evaluation

In this paper, we evaluated the proposed LLF-H method in Section 3 by simulation.

4.1   Simulation Environment

The number of nodes N is $2^1, 2^2, \cdots, 2^{10}$, and 1 source node is included to the nodes. The measurement of the simulation is carried out for each value. As for the delivery cycle C_i of each destination node without any source node, a random delivery cycle from 1 to 10 is given. The maximum number of hops allowable in the LLF-H is $\log_2 N$ because many overlay network construction techniques mention.

According to the existing techniques, sensor data streams having different delivery cycles are delivered as different data streams, and therefore, sensor data is directly delivered from the server. These existing techniques are collectively referred to as Fig. 2. Even if some nodes have the same cycle, the source node sends to other nodes directly. We call these techniques SD (source direct) method and compared with the LLF-H method.

Fig. 7   An example of the LCF method.

In addition, we compared with the approach described in Section 4.1, the longest cycle node in each time receives the sensor data from the source node and redelivers to other destination nodes. When some nodes have the same longest cycle, a specific node of them sends to other nodes. We call this technique LCF (longest cycle first) method. **Figure 7** shows a case by the LCF method in the same example. In the case by the LCF method, $SL = 4.667$ 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 described in Section 3.3. However, a destination node having a small load due to reception during a long cycle sends many sensor data streams to another destination node, and therefore, there is a possibility of the transmission load of the destination node with a long cycle increasing in the case where there are many destination nodes at the same time. In addition, the delivery path is determined on the basis of the number of transmissions and receptions of data, and therefore, the effects of the load distribution are low in an environment where the load cannot be simply represented by the number of transmissions and receptions, for example, $\alpha \neq \beta$.

Moreover, in order to confirm an influence by guaranteeing the maximum number of hops in the LLF-H method, we compared an approach based on load estimation like the LLF-H method but not considering hops. We call this technique the LLF (lowest load first) method. The LLF method is an algorithm that is removed elements related hops from the algorithm of LLF-H method described in Fig. 3.

In this simulation, each method is tried twenty times with the respective numbers of nodes, and the average value of the results is calculated.

4.2   Total System Loads

In the case where the transmission and reception loads per data piece as described in Section 2.4 are $\alpha = \beta = 1$, **Fig. 8** shows the communication load SL of the entirety of the system. The longitudinal axis is the communication load of the entirety of the system, and the lateral axis is the number of destination nodes. In the present simulation, a data set is prepared with the cycle and the order at random, and a terminal having each delivery cycle is added in order. The simulation results are the communication load at the point in time for the number of terminals indicated by the lateral axis, and the greater the number of terminals, the higher the communication load of the entirety of the system.

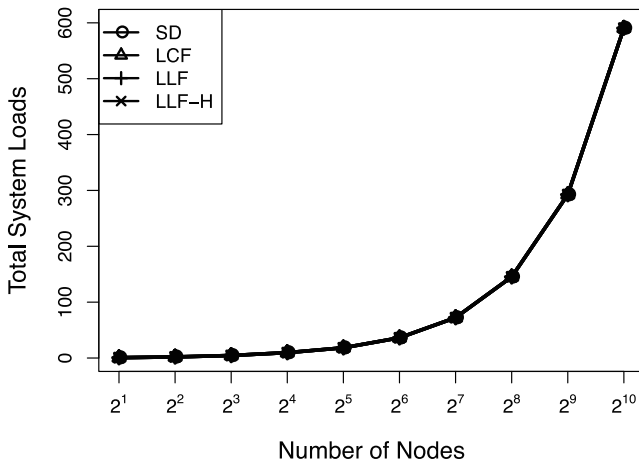According to the SD method, the source node directly delivers

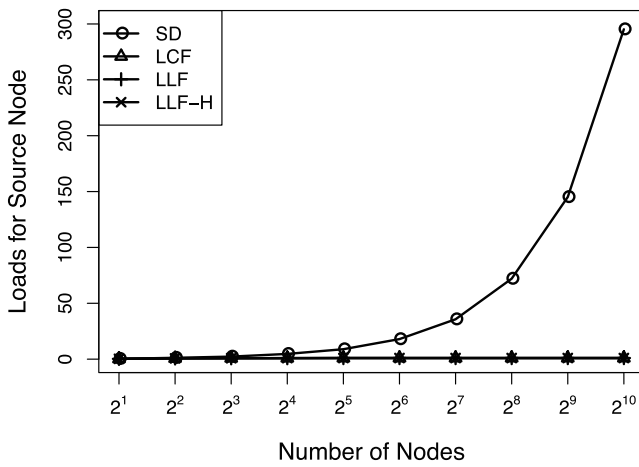**Fig. 8** Total system loads by the number of nodes.

Fig. 9 Loads for source node by the number of nodes.

Fig. 10 Load distribution by the number of nodes.

Fig. 11 Load distribution by $\log_2 \frac{\alpha}{\beta}$.

the sensor data stream, and therefore, there is no redundant transmission or reception of data, and thus, the communication load of the entirety of the system has a minimum value. As can be seen from Fig. 8, the loads of the entirety of the system respectively in the LCF, LLF, LLF-H method are equal to that in the SD method. This is because there is no redundant communication load, for example, the same sensor data are received from some nodes. The value of the load is the same as in the SD method having the minimum value, and therefore, it can be seen that the communication load in the LLF-H method is also kept to a minimum.

4.3 Loads for Source Node

In the case where the transmission and reception loads per data piece are $\alpha = \beta = 1$ as in the previous section, **Fig. 9** shows the communication loads for the source node. The longitudinal axis is the communication load of the source node and the lateral axis indicates the number of nodes.

In the SD method, the source node delivers directly, and therefore, the load on the source node is increased by the number of destination nodes. Meanwhile, the destination nodes redeliver in the other methods, and therefore, the influence of the number of destination nodes is small.

4.4 Load Distribution

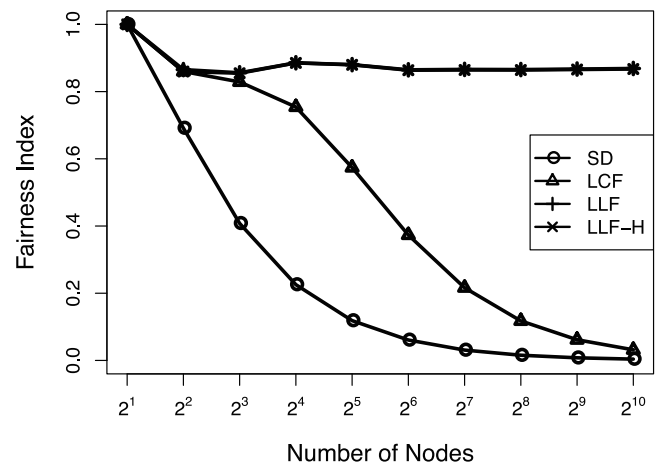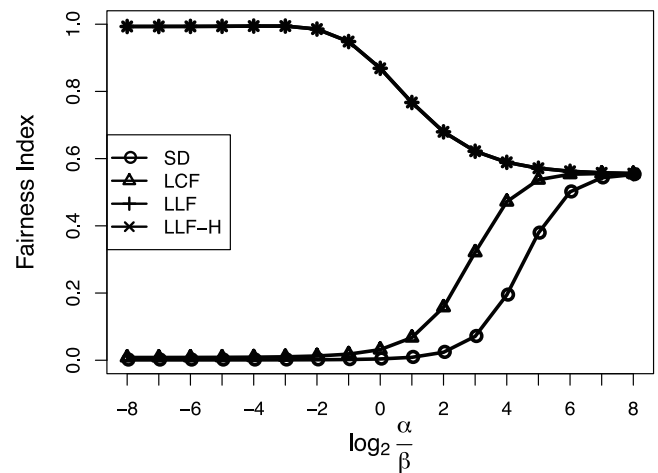In the case where the transmission and reception loads per data

piece are $\alpha = \beta = 1$ as in the previous section, the load dispersion to the destination nodes is shown below.

First, **Fig. 10** shows the results where the longitudinal axis is *FI* and the lateral axis is the number of destination nodes.

In Fig. 10, the greater the number of destination nodes, the smaller the Fairness Index and the more the loads are unbalanced. This is because the greater the number of destination nodes, the longer the longest delivery cycle and the greater the difference in the delivery cycle. In the LLF and LLF-H method, the communication load can be dispersed more than in the LCF method, and in particular, the difference is significant in an environment having a great number of destination nodes. This is because in the LLF and LLF-H method, the destination node having the smallest load at a point in time sends the sensor data stream to another destination node so that the load can be equalized. Meanwhile, in the SD method, the load is concentrated in the source node. In the LCF method, the load concentrates in a destination node having a long cycle when there are many destination nodes at the same time.

Figure 11 shows the *FI* in the case where the transmission and reception loads per data piece α and β as described in Section 2.4 vary. The lateral axis indicates a logarithm of α/β as the ratio of α and β, and in the case where the lateral axis is -2, for example, $\alpha/\beta = 2^{-2} = 1/4$, which means the load for receiving one piece of data is 1/4 of the load sending that the one piece of data. In the
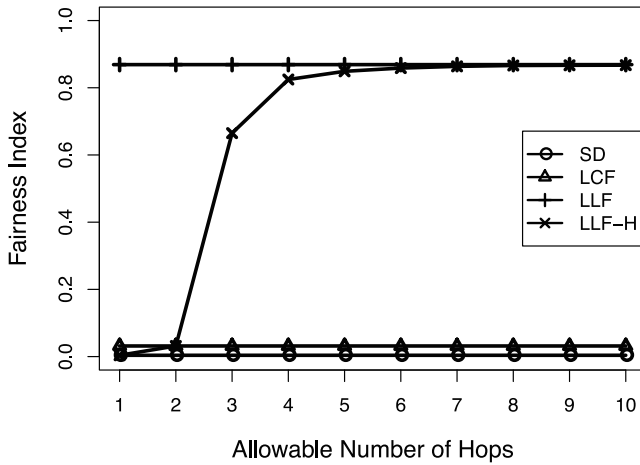
Fig. 12   Load distribution by the number of allowable number of hops.

Fig. 13   The number of hops on each node.

present simulation, there is one source node and $2^{10} = 1024$ destination nodes. In addition, the delivery cycle of each destination node is determined at random between 1 and 10, and the results of ten trials are averaged.

As can be seen in Fig. 11, in the present simulation environment, the *FI* in each technique converges to approximately 0.6 when the reception load α per data piece is maximum and $\log_2 \alpha/\beta$ is great. This is because the number of reception data pieces and the reception load in each destination node are not different between the respective techniques, and the effects due to the difference in the number of transmission data pieces between the respective techniques are relatively small when the reception load per data piece α is maximum. Meanwhile, the load of the destination node protrudes, which lowers the *FI* when the transmission load per data piece β is maximum and $\log_2 \alpha/\beta$ is small, and thus, the effects due to the other destination nodes are small. In reality, it is possible for the ratio of the reception and transmission loads to be approximately 2 at the greatest. In a range of $-1 \leq \log_2 \alpha/\beta \leq 1$ where the ratio is 2, the *FI* is the greatest in the LLF and LLF-H method where it can be seen that the load is dispersed.

Figure 12 shows the *FI* by the allowable number of hops. The lateral axis is the maximum number of hops which is allowable in the LLF-H method. In this simulation, the allowable number of hops is changed from 1 to 10. The number of nodes is $2^{10} = 1024$.

As can be seen in Fig. 12, the LLF-H method achieves approximate *FI* with the LLF method when the allowable number of hops is 7. In the cases of less hops than 7, the *FI* drops. In this simulation, the number of nodes is 2^{10} and the number of hops mentioned in many overlay network construction techniques is $\log_2 2^{10} = 10$. Thus, the LLF-H seems to achieve the valid number of hops with approximate *FI* by the LLF method.

4.5   The Number of Hops

Figure 13 shows the number of hops until each node receives data at the time of the lowest common multiple of delivery cycles. All nodes receive data at the time of the lowest common multiple. In this simulation, the number of nodes is $2^6 = 64$ and delivery cycle of each node is determined at random between 1 to 10. The lateral axis shows each node as ID $0 \sim 63$. The longitudinal axis
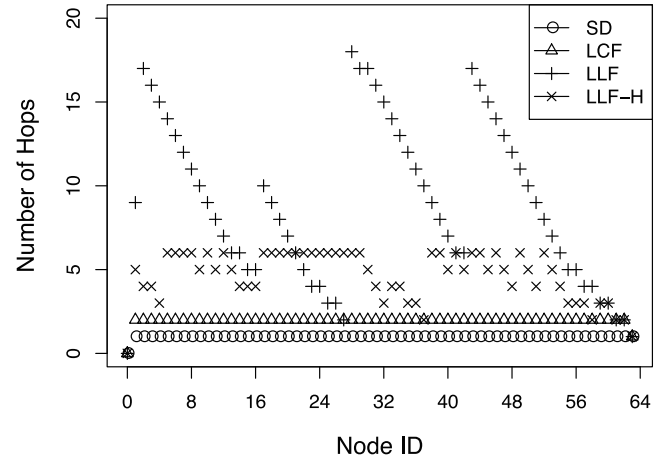
is the number of hops until each node receives data at the time of the lowest common multiple. The node IDs in the lateral axis are sorted from the shorter cycle node. The node with ID = 0 is the source node. In this simulation, the allowable number of hops in the LLF-H is 6.

As can be seen in Fig. 13, the number of hops of the source node (ID = 0) is always 0 in all methods. The number of hops of the longest cycle node at the time of the lowest common multiple (ID = 63) is 1 in all methods. In the SD method, the source node delivers to all destination nodes. The numbers of hops of the nodes without the source node are 1. In the LCF method, the longest cycle node (ID = 63) sends to all destination nodes. The numbers of hops of the nodes without the source node (ID = 0) and the longest cycle node (ID = 63) are 2. In the LLF method, the delivery paths are constructed without considering the number of hops and the maximum number of hops is 18. In this simulation, the number of nodes is 2^6 and the number of hops seems large comparing with $\log_2 n = \log_2 2^6 = 6$. Since the number of hops is increased by the number of nodes, the number of hops in the LLF method is large and a delivery delay happens. On the other hand, in the LLF-H method, another low load node sends data if the number of hops seems over the specified number 6. Thus, the number of hops of each node is less than 6.

4.6   Communication Loads of Each Node

Figure 14 shows the communication loads of the source node and the destination nodes in the case where the number of destination nodes is ten and the cycles are $1, 2, \cdots, 16$. The longitudinal axis is the communication load of each node, and the lateral axis indicates the ID of nodes where the ID of the source node is 0. The transmission and reception loads per data piece are $\alpha = \beta = 1$ as in the previous section.

In the SD method, the source node directly delivers the sensor data stream, and therefore, the load concentrates on the source node of which the ID is 0. The existing P2P techniques described in Section 1 perform like the SD method in this environment where there are no same cycles in destination nodes. Thus, the load extremely concentrates on the source node in the existing P2P techniques. In the LCF method, destination nodes having a long cycle send the sensor data stream with priority, and therefore
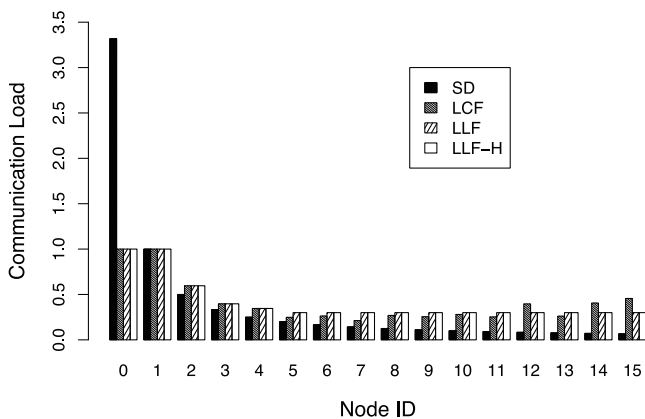
Fig. 14 Communication loads of each node.

have a load greater than that of the destination nodes of which the cycle is close to the median. Meanwhile, in the LLF and LLF-H method, loads can be equalized between the destination nodes having a delivery cycle at a certain level or higher as compared to the LCF method.

5. Related Work

Various techniques for dispersing the communication load in the delivery of streams have been studied.

A P2P stream delivery technique according to which a stream is delivered using a P2P technology for sending and receiving data between terminals having no servers in between has been proposed in order to disperse the communication load among the terminals [1], [2], [3], [4], [5]. The P2P stream delivery technique is divided into a pull type technique and a push type technique. In a pull type technique, such as PPLive [*1], DONet [1], and SopCast [*2], the reception terminal that receives data requests data from another terminal and acquires it. Though communication is carried out in order for the reception terminal to find a terminal that has data that has not yet been received by the reception terminal, no such redundant communication whereby the data that has already been received by the reception terminal is again requested is carried out. In a push type technique, such as AnySee, data is sent from the transmission terminal that sends data to another terminal [2]. Though the communication is carried out in order for the transmission terminal to find a terminal that has not yet received data received by the transmission terminal, no such redundant communication is carried out whereby the data that has already been received by the reception terminal is distributed again. A technique where a pull type and a push type are combined, such as PRIME, has been proposed [3].

In P2P stream delivery techniques, a case where the same data stream is delivered to a number of terminals is assumed. In the delivery of the sensor data streams, however, there are cases where a data stream of the same sensor having different delivery cycles is delivered. In this case, sensor data streams having different delivery cycles are delivered as different data streams. Thus, as described in the evaluation of Section 4.6 or like Section 4.3, the communication load on the source node cannot be efficiently dis-

persed.

Several techniques for preventing the communication load from being concentrated on a particular terminal by constructing a data delivery path, which is referred to as a multicast tree, in advance so that a data stream is delivered have been proposed [6], [7], [8], [9], [10]. In the ZIGZAG method, a multicast tree is constructed by clusters that are collections of terminals [7]. The number of clusters included in each depth of a multicast tree is made the same, and thus, the load is 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 communication load can be prevented from concentrating on a particular terminal as compared to ZIGZAG by taking the communication delay between terminals into consideration in the case where the physical network structure can be understood [8]. The implementability of the MSMT/MBST method was poor because it is necessary to understand all the network structures between the terminals concerning stream delivery. In LAC (locality aware clustering), a load dispersion higher than that in ZIGZAG is achieved by taking into consideration the communication delay between only some terminals, though the physical network structure cannot be understood [9].

In the case where sensor data streams have different delivery cycles as in the above-described P2P stream delivery technique, sensor data streams having different delivery cycles are delivered as different data streams. Thus, described in the evaluation of Section 4.6 or like Section 4.3, the communication load on the source node cannot be efficiently dispersed.

6. Conclusion

The present study has proposed the LLF-H method according to which the communication load is dispersed in a P2P model in the case where sensor data streams having different delivery cycles are delivered. In the proposed method, sensor data delivered at the same time and included in different sensor data streams is redelivered so that the communication load is dispersed. In addition, the number of hops to destination nodes is guaranteed. As a result of the evaluation, we confirmed that the communication load on the entirety of the system could be dispersed to the respective terminals satisfying a limitation for the number of hops.

In the future, we will study a technique to disperse the communication load by relay nodes when there are a number of source nodes. In addition, a determined delivery path in the proposed method is updated when the state of nodes changes. Therefore, we will study a technique to reduce an influence in an environment where the state of nodes changes frequently.

---

gramme (SCOPE) of the Ministry of Internal Affairs and Communications, Japan.

References

[1] Zhang, X., Liu, J., Li, B. and Yum, T.-S.P.: CoolStreaming/DONet: A Data-Driven Overlay Network for Peer-to-Peer Live Media Streaming, *Proc. 24th Annual Joint Conference of the IEEE Computer and Communications Societies* (*INFOCOM 2005*), Vol.3, pp.2102–2111 (2005).

[2] Liao, X., Jin, H., Liu, Y., Ni, L.M. and Deng, D.: AnySee: Peer-to-Peer Live Streaming, *Proc. 25th IEEE International Conference on Computer Communications* (*INFOCOM 2006*), pp.1–10 (2006).

[3] Magharei, N. and Rejaie, R.: PRIME: Peer-to-Peer Receiver-Driven Mesh-Based Streaming, *Proc. 26th IEEE International Conference on Computer Communications* (*INFOCOM 2007*), pp.1415–1423 (2007).

[4] Yu, L., Liao, X., Jin, H. and Jiang, W.: Integrated Buffering Schemes for P2P VoD Services, *Peer-to-Peer Networking and Applications*, Vol.4, No.1, pp.63–74 (2011).

[5] Sakashita, S., Yoshihisa, T., Hara, T. and Nishio, S.: A Data Reception Method to Reduce Interruption Time in P2P Streaming Environments, *Proc. 13th International Conference on Network-Based Information Systems* (*NBiS*), pp.166–172 (2010).

[6] Banerjee, S., Bhattacharjee, B. and Kommareddy, C.: Scalable Application Layer Multicast, *Proc. ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (*SIGCOMM 2002*), pp.205–217 (2002).

[7] Tran, D.A., Hua, K.A. and Do, T.: ZIGZAG: An Efficient Peer-to-Peer Scheme for Media Streaming, *Proc. 22nd Annual Joint Conference of the IEEE Computer and Communications Societies* (*INFOCOM 2003*), Vol.2, pp.1283–1292 (2003).

[8] Jin, X., Yiu, W.-P.K., Chan, S.-H.G. and Wang, Y.: On Maximizing Tree Bandwidth for Topology-Aware Peer-to-Peer Streaming, *IEEE Transactions on Multimedia*, Vol.9, No.8, pp.1580–1592 (2007).

[9] Silawarawet, K. and Nupairoj, N.: 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] Le, T.A. and Nguyen, H.: Application-Aware Cost Function and Its Performance Evaluation over Scalable Video Conferencing Services on Heterogeneous Networks, *Proc. IEEE Wireless Communications and Networking Conference: Mobile and Wireless Networks* (*IEEE WCNC 2012 Track 3 Mobile and Wireless*), pp.2185–2190 (2012).

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, he was a specially appointed researcher at Osaka University. Since April 2013, he has been a Ph.D. researcher at Kobe University. His research interests include distributed processing systems. He is a member of IPSJ.

Yoshimasa Ishi received his Bachelor's degrees from Kyoto Institute of Technology, Japan, in 2004 and his Master's degrees from Osaka University, Japan, in 2006, respectively. From 2006 to 2008, he was a specially appointed researcher of Cybermedia Center, Osaka University. From 2008 to 2012, he was a specially appointed researcher of Graduate School of Information Science and Technology, Osaka University. Since April 2012, he has been a specially appointed researcher of Cybermedia Center, Osaka University. His research interests include technologies for distributed network systems and its development. He is a member of IPSJ.

Tomoki Yoshihisa received his 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 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 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 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 IPSJ and IEEE.