A Distributed Delivery System for Sensor Data Streams with Different Delivery Cycles

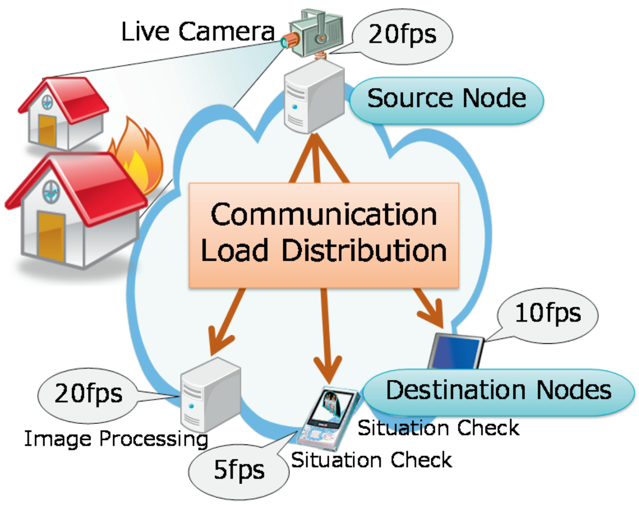
Tomoki YOSHIHISA and Yuuichi TERANISHI

Due to the development of information and communications technology, sensor data stream delivery systems which cyclically deliver sensor data, such as live cameras or environmental sensors for fire discovery. We can realize short delay and low loss delivery systems by distributing the communication loads of sensor data stream delivery systems and this leads to our safe and trusted lives. In this research, we research and develop a distributed sensor data stream delivery system for the cases of delivering data with different cycles.

1 Introduction

With developments in information technology, sensors such as live cameras and environmental sensors are being installed at multiple points. Conventionally, these sensors were each used for one purpose, but the sensor data obtained from these sensors can be applied to diverse application purposes, so there is increasing demand for shared usage over different purposes. In this context, in recent years, the focus has been on the sensor data stream delivery systems that deliver sensor data periodically to multiple nodes with different application purposes. Furthermore, a group of sensor data series generated consecutively from a sensor is known as a sensor data stream. For example, we can imagine that live camera images are constantly delivered to the computing node that carries out image processing for fire detection and the same live camera images are delivered to the user's smartphone node that is to display the images to check the state of the place that is being photographed. In Figure 1, the computing node connected to the live camera delivers the images, and may also have the delivery function in the live camera itself. More examples include weather data of environmental sensors that is constantly delivered to the computing node for prediction of abnormal weather, and weather data of the same environmental sensors that is delivered to a car navigation node for visualization, for driving while checking the weather at the travel destination.

In sensor data stream delivery, when there is a concentration of communication load such that much data is delivered in a short time to a particular node, and when the communication buffer for temporary storage of the delivered data is not sufficient in the node, the delivered data cannot be received and data is lost. The load applied on the node along with data send and receipt is referred to as communication load here, as described later in Subsection **2.4**. If data is lost, such as if images of the live camera, before and after the occurrence of a fire, are lost, fire detection is delayed and safety is threatened, the quality of service that should be met is not achieved, and the delivery system cannot be used in a reliable manner. Even in cases of sufficient communication buffers, processes such as image processing and analysis for much data take time, and there is a long delay for acquisition of processing results. Weather data analysis takes time, resulting in delayed prediction of abnormal weather, leading to threatened safety. For long delays, the delivery system cannot be used similarly in a reliable manner. For long delays, the time



fps (frames per seconds): # of images (frames) per 1 sec.

Fig. 1 Application targets of this study

taken for storage in the communication buffer is also long, causing loss. For improved processing speed, and provision of sufficient communication buffers, financial cost is involved. Due to these reasons, to achieve sensor data stream delivery that involves low loss and short delays, leading to our safe and untroubled living conditions, various studies have been carried out for distributed communication load[1]–[4]. Distributed communication load can reduce the possibility that load is concentrated in one part even for multiple delivery destinations, and that operations of the entire system are not maintained normally.

In these studies, when delivering the same sensor data to multiple nodes, the node that fetches sensor data from the sensor (source node) does not transmit data directly, and the node of the delivery destination that receives sensor data (destination node) further re-delivers sensor data to other destination nodes. The destination node, by re-delivering sensor data, distributes the communication load that is concentrated in the source node. The subject of these studies was the delivery system that delivers sensor data in the same cycles to all destination nodes, but with an increase in the application targets of sensor data, the delivery system that handles different delivery cycles is being used. Below is an example.

- In the example of the live camera here above, in processes involving detection of phenomena that is photographed as images, such as fire detection, missed detections can be reduced by increasing the images that can be used for detection. To detect a fire without mistakes, deliver the image in short cycles to the computing node, and increase the number of images that can be used per unit time. On the other hand, the processing capacity of smartphones is lower than the computing nodes such as mentioned above, which carry out image processing constantly, and when delivery is in the same cycle, there is a long delay until display of the image and there are times when there is no display. For this reason, in user smartphone nodes that check the state of the location captured by the same live camera, deliver in cycles longer than for the above computing node, but to the extent where the state can be checked.
- In the example of the environmental sensor here above, in processes involving analysis of weather data, such as prediction of abnormal weather, the analysis precision can be improved through increase in the data that can be used for analysis. To predict abnormal weather accurately, increase the quantity of data that can be used per unit time by delivering weather data in short cycles to the computing node. On the other hand, the communication band of the car navigation system that is used during movement is smaller than the abovementioned computing node and when delivery is in the same cycle, there is a long delay until receipt of the weather data and there are times when there is no receipt. For this reason, in the car navigation node that displays an image for checking weather at the travel destination using the same environmental sensor, deliver in cycles longer than for the above computing node, but to the extent where weather can be checked.

By ensuring that the source node delivers sensor data to all destination nodes in its shortest possible cycles and each delivery node uses only the sensor data that it needs, it is possible to achieve delivery cycles that differ artificially and apply the method used hitherto. However, as this system results in delivering redundant sensor data not to be used in the destination node, the communication load will be more compared to when delivering just the sensor data that is to be used. And if the communication load increases, there might be data losses or longer delays, as described above, and hence the delivery system cannot be used in a reliable manner.

So in this study, we have researched and developed a distributed delivery system for sensor data streams with different delivery cycles[5][6]. In delivery systems for sensor data streams with different delivery cycles, the research on distributed communication load is unprecedented and the novelty is high. Moreover, with an increase in the application targets of sensor data, there is a sudden increase in delivery systems that handle different delivery cycles, and the gravity is very high.

Explained in Section **2** below is the model of the distributed delivery system for sensor data streams with different delivery cycles as well as problem formulation. The basic ideas of this study are also explained in Section **2**. In this study, each model proposes a few methods for communication load distribution and evaluates them using computer simulation or actual measurement. The result of evaluation is that the distribution of the communication load could be confirmed by using the proposed methods. Study contents of each model are explained in Sections **3** to **5**. Finally, this paper concludes in Section **6**.

2 Model of distributed delivery system for sensor data streams with different delivery cycles

For research and development of a distributed delivery system for sensor data streams with different delivery cycles, we carried out modeling and problem formulation. This section first explains these research contents. Furthermore, it explains the basic ideas of the proposed communication load distribution methods.

2.1 Assumed environment

The node that can fetch sensor data from the sensor and deliver it is referred to as the source node. The source node fetches sensor data in cycle *s* and is connected to the information network, such as the Internet or Intranet. The destination node is connected to the same information network as the source node, and can communicate with the source node. Each node can further deliver the received sensor data to other nodes. The destination node first discovers the source node that can deliver the sensor data that is to be used, from information that has been published using methods such as a website homepage or RSS etc., and requests sensor data stream delivery from the source node in the desired delivery cycle *d*. The source node cannot deliver the sensor data that it has not fetched yet, and thus *d* is a multiple of *s*. Furthermore, if the source node receives the delivery request, it considers the current delivery destination and determines the delivery route. When there is a change in the delivery route, it gives notice of such change in the delivery destination to the node chosen as the new destination. The delivery route changes every time the source node receives the sensor data stream delivery request.

For example, in the example of the Section **1** live camera, the node connected to the live camera is the source node, and if it fetches 20 images every second from this live camera, s=1/20=0.05. The destination node will be the computing node that processes images (destination node 1) and the smartphone node that displays images (destination node 2), and if destination node 1 requests 20 images every second, which is the same as the source node, d=0.05, and if the destination node 2 requests 5 images every second, d=1/5=0.2. The destination node requests image delivery by discovering the live camera to be used, from the live camera site on the Internet, etc.

2.2 System model

Through the division of the distributed sensor data stream delivery system by the connection status of the source node, relay node and destination node, a discussion is possible according to the connection status. The connection status indicates the communication link of the application layer. The relay node is neither a source node nor is it a destination node, but refers to a node that is involved in the distribution of the communication load by relaying sensor data streams. When these are used without distinction, they are simply described as nodes. For example, the router within the information network and wireless access point are referred to as relay nodes. Shown here below is the node connection status in $l/m/n$ format. l is the number of source nodes, m is the number of relay nodes, and n is the number of destination nodes.

- 1/0/1 model: It contains 1 source node *S*, and 1 destination node *D*. There are only 2 nodes, and the connection status is as in Fig. 2. In Figures 2 to 5, each node indicates a terminal, and the values described near the node are the sensor data acquisition cycle *s* of Node *S*, and the delivery cycle *d* requested by Node *D*. The branch that extends from *S* to *D* is written with a dotted line, not a solid line, because a communication link between them is not always chosen as a delivery route of sensor data. In the 1/0/1 model shown in Fig. 2, there could be only one delivery route through which *S* directly delivers sensor data to *D*.
- 1/M/1 model: Between 1 source node and 1 destination node, M relay nodes I_m(m=1,…, M) are provided. When the delivery source delivers sensor data directly to the delivery destination without using the relay node, the delivery route is the same as the 1//0/1 model.
- 1/0/N model: Shown in Fig. 3 is the connection status of 1/0/N model. N destination nodes D_n(n=1,…, N) exist. The delivery cycles to D_n are shown in d_n. By re-delivering sensor data between destination nodes, it is possible to distribute the communication load applied on the source node. By deciding the delivery route such that the communication load does not concentrate on a particular destination node, it is also possible to distribute the communication load of the destination node.
- 1/M/N model: Shown in Fig. 4 is the connection status of 1/M/N model. By using the relay node, the communication load on the source node and

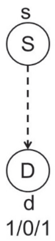


Fig. 2 Example of the 1/0/1 model

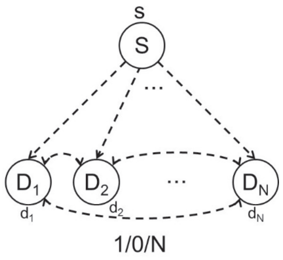


Fig. 3 Example of the 1/0/N model

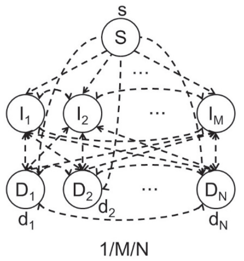


Fig. 4 Example of the 1/M/N model

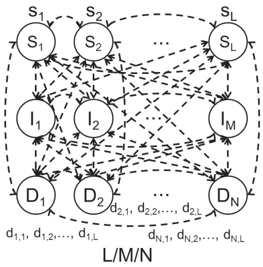


Fig. 5 Example of the L/M/N model

destination node in 1/0/*N* model can be distributed in relay nodes. However, increase in the communication hop count between the source node and the destination node leads to longer delays.

- *L*/*M*/*N* model: Shown in Fig. 5 is the connection status of the *L*/*M*/*N* model. L source nodes S_l((l=1,…, L) exist. The collection cycle of the sensor data received by the destination node D_n (n=1,…, N) from the source node S_l (l=1,…, L), is shown as $d_{n,l}$ (l=1,…, L). In the *L*/*M*/*N* model, it was possible to distribute the load by performing sensor data relay between the source nodes.
- Other models: Models such as *L*/0/1, *L*/*M*/1, *L*/0/*M* exist in addition to the models explained here above.

These models are dependent on the number of source nodes and number of destination nodes, as well as network configuration.

2.3 Objective functions

The communication load of node *N* is represented as *C(N)*. The communication load *SL* of the entire sensor data stream delivery system is considered to be the total of the communication load of each node, and is represented as the following equation.

$$SL = \sum_{l=1}^{L} C(S_l) + \sum_{m=1}^{M} C(I_m) + \sum_{n=1}^{N} C(D_n) \tag{1}$$

Moreover, *FI* (Fairness Index) is often used as the index of load distribution. In this study as well, we have used the *FI* as applied in the following equation, as the load distribution index.

$$FI = \frac{\left(\sum_{l=1}^{L} C(S_l) + \sum_{m=1}^{M} C(I_m) + \sum_{n=1}^{N} C(D_n) \right)^2}{(L+M+N)\left(\sum_{l=1}^{L} C(S_l)^2 + \sum_{m=1}^{M} C(I_m)^2 + \sum_{n=1}^{N} C(D_n)^2 \right)} \tag{2}$$

FI shows that it is fairer closer to 1, and for *FI*=1, $C(S_1)= \cdots =C(SL)=C(I_1)=\cdots=C(I_M) \;\; =C(D_1)=\cdots=C(D_N)$. Moreover, the communication hop count that accompanies data delivery from the source node *P* to the destination node *Q* is taken as *H*(*P*,*Q*). This study aims at preserving the soundness of the entire system operations, by ensuring a fairly shared communication load and eliminating any bottlenecks. Therefore, the fairness in sharing the communication load to be achieved while suppressing the communication load of the entire system was taken as the objective function. Moreover, there have also been demands to reduce the communication hop count, and shorten the delays on delivery. Thus the objective function is considered to be as here below.

$$\text{minimize}\, SL, 1-FI, \max_{p\in P,\, q\in Q} H(p,q) \tag{3}$$

P is expressed as the set of source nodes; *Q* is expressed as the set of destination nodes. This study determines the sensor data to be delivered by each node according to the objective function here above.

2.4 Definition of the communication load

The specific communication load of each node depends on the performance of the node, and is proportionate to the data count per unit time of send and receive. The data count to be received per unit time by node *N* is *R(N)*, the data count to be sent per unit time is *T(N)*, and the steady communication load is *S(N)*. The communication load of *N* is assigned the following formula.

$$C(N) = rR(N) + tT(N) + S(N) \tag{4}$$

r and *t* are weighting factors of the communication load that accompanies receipt and the communicating load that accompanies send. In this study, since the result of actual measurement evaluation confirms that *S(N)* is much smaller than *R(N)* and *T(N)*, hereafter *S(N)*=0. Moreover, unless otherwise stated, *r*=*t*=1. The example in case of direct delivery by the source node is shown in Fig. 6 (a). The number displayed near the branch shows the sensor data count that is to be delivered per unit time. In this example, $C(S)$=1+1/2+1/2+1/4=2.25, $C(D_1)$=1, $C(D_2)$=0.5, $C(D_3)$=0.5, $C(D_4)$=0.25 and *SL*=4.667, *FI*=0.617, where the maximum hop count is 1.

2.5 Basic idea

In the proposed method, the communication load is distributed by re-delivering sensor data of the same clock time as included in different sensor data streams. For example, Table 1 shows the delivery cycles (Cycle) to destination nodes D_1 to D_4 and the sensor data to be delivered is indicated as . D_1 has Cycle 1 in which the source node receives all sensor data fetched from the sensor. D_2 has Cycle 2, and sensor data is received every 2 clock times. For example, clock time 6 sensor data is included in sensor data streams of all of Cycles 1, 2, 3, and 4. For this reason, with the delivery of sensor data of clock time 6 to other re-generating nodes by the delivering destination nodes that receive clock time 6 sensor data, there is no direct delivery from the server, and sensor data can be redelivered. Figure 6 (b) shows an example in which the delivery cycle is the same as Fig. 6 (a) and the communication load is distributed by changing the delivery route. In the example of Fig. 6 (b), the destination nodes having long delivery cycles redeliver data to other destination nodes that receive the same data. In each re-generating node, only the sensor data that is to be used is delivered, and the total of sensor data count per unit time to be received is equal to the reciprocal number of the delivery cycles. In this case, $C(S)$=1, $C(D_1)$=1, $C(D_2)$= 0.833, $C(D_3)$=1, $C(D_4)$=0.833 and *SL*=4.16, *FI*=0.992, where the maximum hop count is 3. It is evident that the communication load is better distributed compared to the previously described case in which the source node directly delivers data (*FI*=0.617).

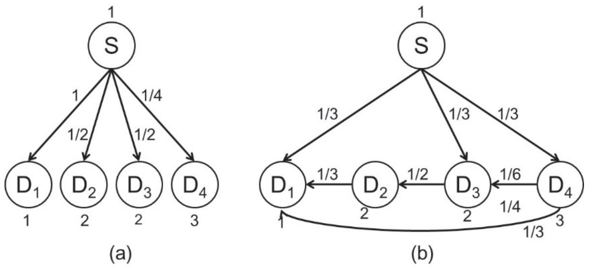


Fig. 6 Communication load and basic idea

3 Communication load distribution method in 1/0/N model

We propose two methods for distributing the communication load in the 1/0/N model[7].

3.1 LCF (Longest Cycle First) method

In the destination node with a long delivery cycle, the amount of sensor data being received per unit time is less, and the communication load is less compared to the node with a short delivery cycle. For this reason, it may be possible to distribute the communication load by delivering sensor data to other destination nodes to increase the communication load to these nodes, while reducing the communication load of the source node. Here, in the LCF (Longest Cycle First) method, the destination node with a long delivery cycle redelivers the sensor data on a priority basis to distribute the communication load. For example, as in the case of Table 1, the delivery route will be what is shown in Fig. 6 (b). In the LCF method, the delivery route can be built from only that information of the delivery cycle which each destination node will request, and installation is easy.

Table 1 Clock time for delivery of data in the connection state of Fig. 6

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

3.2 LLF (Lowest Load First) method

In the LCF method, load distribution cannot be conducted effectively, since the communication load is not taken into consideration. For example, in the destination node with a long delivery cycle, the transmission load will increase when there are several time-synchronized delivery destinations, since that node delivers data to many other destination nodes. Here, in the LLF (Lowest Load First) method, the communication load is estimated from the delivery cycle of each node, and from among the nodes receiving the time-synchronized sensor data, the node with the smallest communication load is chosen as the delivery source, thus distributing the communication load. In the LLF method in which the load is estimated, the load can be more evenly distributed compared to the LCF method.

3.3 Evaluation

The performance of the proposed methods was evaluated through computer simulations. Figure 7 shows the fairness in the 1/0/N model, when the delivery cycle is randomly assigned between 1 and 20. The vertical axis is the value of FI, and the horizontal axis is the *N* number of destination nodes. SD is the method used for comparison, in which the source node directly delivers data to each destination node. The more destination nodes, the lesser the FI, and deviations will occur in the communication load. This is because, the more destination nodes, the longer the longest delivery cycle, and the wider the differences between delivery cycles. The communication load is much better distributed in the LLF method compared to the LCF method, and, especially in an environment where there are several destination nodes, the difference is huge. This is because in the LLF method, the delivery destination with the lowest load each time transmits to other delivery destinations, thereby distributing the communication load. On the other hand, in the SD method, the communication load is concentrated in the source node, and in the LCF method when there are several time-synchronized delivery destinations, the load is concentrated in the delivery destination with a long cycle. The communication load *SL* of the overall delivery system was evaluated, and it was confirmed that there are no differences among these methods.

From these evaluations, in this research we confirmed that in comparison with direct delivery, the communication load can be better distributed using the proposed LCF method and LLF method.

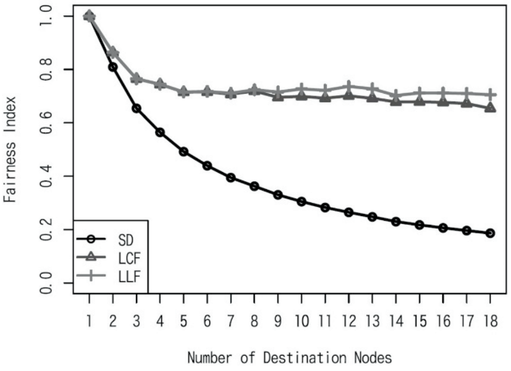


Fig. 7 Evaluation of fairness in 1/0/N model

4 Communication load distribution method in L/0/N model

In this research, evaluations were conducted by adapting the LLF method, the validity of which was confirmed using the 1/0/N model, to the L/0/N model[8]. Moreover, taking into consideration the communication hop count, it was expanded and implemented, and measured evaluations were conducted using testbeds[9].

4.1 M-LLF (Multi-stream LLF) method

In the LLF method, first, the communication load of each node is estimated from the delivery cycle etc., and then the undecided delivery route is decided for each time and for each destination node. That delivery route is to selected, through which the communication load estimated for each time can be transmitted from the smallest node, and after all the delivery routes are decided while being reflected in the results of communication load estimation, the delivery routes of each final time will be notified to all nodes. In the L/0/N model, that has multiple source nodes, there are also multiple sensor data streams. In the M-LLF method in which the LLF method has been adapted to the multiple sensor data streams, the communication load estimated by the delivery of each sensor data stream is accumulated. With this, even in cases where there are other sensor data streams, the communication load can be evenly distributed.

Results of the evaluation through computer simulation confirmed that the fairness in sharing the communication load can be improved, without increasing the communication load SL of the overall delivery system, by using the M-LLF method. Because of space limitations, the details of the evaluation results are omitted here. Please refer to the Reference [8] for the details.

4.2 LLF-H (LLF considering Hops) method

In the M-LLF method, the focus is on the distribution of the communication load, and delivery delays are not taken into consideration. For this reason, the number of communication hops increases, leading to longer delay. Here in the LLF-H method, delivery delays are controlled by placing an upper limit on the number of hops for redelivering sensor data. Specifically, in the procedure for selecting the node with the least communication load in the M-LLF method, it is judged whether the number of communication hops that would occur when redelivery is received from a candidate destination node would not exceed the predetermined upper limit. In case the number of communication hops exceeds its upper limit, the same judgment will be made with the next candidate, and when there is no destination node that meets the conditions, data will be received from the destination node having the longest delivery cycle from all the delivery cycles.

4.3 Evaluation using PIAX testbed

Since an application layer network can be built easily by using PIAX in which NICT has provided a testbed, it is better suited for the proposed methods[10]. Here, in this research, the LLF-H method was evaluated using a PIAX testbed. Assuming the environment sensor etc., with the data size as 1024 bytes, sensor data was acquired at 50-millisecond intervals.

Figure 8 shows the measured value of delivery delay. The vertical axis is the delay time, and the horizontal axis is the upper limit on the number of communication hops. Plotting is being done when there are several different destination nodes. In most cases, it is found that there is least delivery delay when the upper limit on the number of communication hops is about 3 to 4. This is because, in the case where the upper limit on the number of communication hops is 2, the communication load is concentrated, since a particular destination node where the sensor data is directly received from the source node redelivers the data to all other destination nodes.

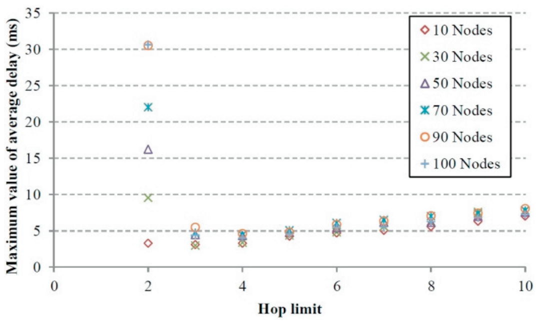


Fig. 8 Results of evaluation using PIAX testbed

5 Communication load distribution method in L/M/N model

In the L/M/N model, the more relay nodes, the bigger the communication load from the transmission and receiving of the relay nodes, resulting in an increase in the communication load of the overall delivery system. For this reason, mainly as a countermeasure against failure, this model will be utilized when using relay nodes. For example, for building a delivery route, the components as shown in Fig. 9 are considered. Figure 9 shows that temperature sensor S_1 and live camera S_2, which are the two source nodes, deliver the sensor data stream to the destination nodes D_1 to D_4 via the relay nodes N_1 to N_3. The portion enclosed by a square in each node indicates the delivery cycle of the sensor data stream, and S_1 delivers the data stream *a* of temperature data in the delivery cycle *s*, and S_2 similarly delivers the data stream *b* of pictures in the delivery cycle *s*. S_1 is 0 since data stream *b* is not delivered. The relay node relays the data stream of the optional delivery cycle, and delivers to other relay nodes or delivery destinations. The destination node receives several data streams, for example, D_2 receives data stream *a* in delivery cycle 2, and also data stream *b* in delivery cycle 2. In a method such as the LLF-H method, where only communication load distribution and delays are taken into consideration, improvements in fault tolerance using relay nodes cannot be expected. Here in this research, we prepare a method by which the communication load of relay nodes is autonomously distributed by using a distributed hashing[11].

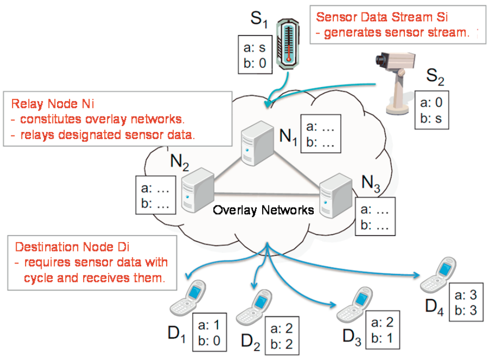


Fig. 9 Example of delivery route in L/M/N model

5.1 Communication load dispersion method using a distributed hashing

In order for a relay node to autonomously build a delivery route, nodes are placed in the hash space, and the sensor data to be delivered is decided based on distributed hashing. At this time, in deciding the relay node that will deliver, we are considering use of distributed hashing in which “Sensor data stream”, “Sensor data stream and Cycle”, and “Sensor data stream and Time” are utilized. For example, using the distributed hashing of sensor data stream, the relay node that will handle each sensor data stream is decided, and sensor data is received from the source node and delivered to the destination node. However, depending on the amount of sensor data of each sensor data stream and the number of destination nodes, deviations will occur in the communication load. Here, this research proposes a method wherein the relay nodes will be grouped according to the sensor data stream and cycle, and the relay nodes that will deliver each time in every cycle group will be decided. By deciding the relay nodes for each time, the process will be finely distributed, and concentration of communication load at a particular time and in a particular relay node is controlled.

Figure 10 shows the results of how the maximum communication load was obtained through computer simulations, by randomly assigning the delivery cycle from 1 to 100, and changing the number of relay nodes. Since distributed hashing is used, it is found that the communication load is distributed in each method, and there is no major difference in the maximum communication load. On the other hand, Figure 11 shows the fairness. The vertical axis is FI. In the case when there are several relay nodes, since the relay nodes not handling anything are increasing, the deviations in the communication load become large when the method of using distributed hashing of sensor data stream and the method of using distributed hashing of cycle are used. On the other hand, in the proposed method where distributed hashing of cycle and time is used, it is found that FI becomes high, and the communication load is distributed even in the case where there are several relay nodes.

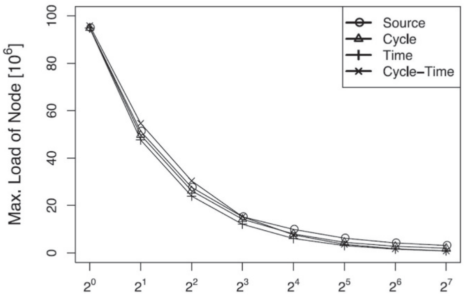


Fig. 10 Maximum communication load in the method of using distributed hashing

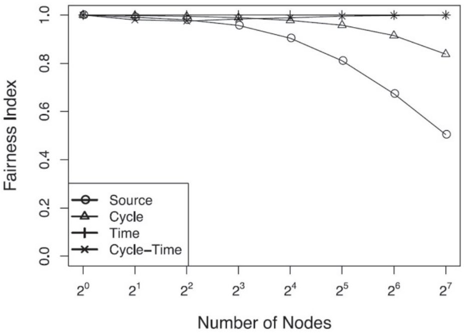


Fig. 11 Fairness in the method of using distributed hashing

5.2 Fault tolerance improvement method using distributed hashing

In the abovementioned method, a relay node is placed in the hash space for each sensor data stream, and the hash space for each delivery cycle is divided and grouped, so that the shorter the delivery cycle groups, the more numerous the relay nodes. When a relay node is not included in a particular group of delivery cycles, the group previous to it will handle the delivery. In this research, we considered a method for making the relay node of each time redundant using a Successor List, in order to improve the fault tolerance when the relay node is disconnected from the network or powered off, etc. Even in the existing distributed hashing method, Chord[12], etc., redundancy is being achieved by using the redundant node information (Successor List) for link structure maintenance. In this research, the length of the Successor List (the number of relay nodes to be made redundant) is set beforehand, and the primary relay node relays the sensor data to all Successors. In case the primary relay node fails, a part of the restructuring process of the delivery route can be streamlined, since the Successor on the distributed hashing will become the next relay node. The results of the evaluation confirmed that when using a Successor List, the communication load increases to a certain degree, but the probability of continuing with the delivery even in the event of the failure of the relay node also increases. Because of space limitations, the details of the evaluation results are omitted here. Please refer to the Reference [13] for the details.

6 Conclusion

This research performed research and development on the distributed delivery system for sensor data streams with different delivery cycles. Specifically, a model of the delivery system was created and the problems formulated, and several methods for distributing the communication load were proposed. The results of the evaluations conducted through computer simulations and of the measured evaluations confirmed that the communication load can be distributed by using the proposed methods.

For the future, we are considering research and development on the distribution method of communication load and process load in the distributed processing-type sensor data stream delivery system that will also stream deliver the process results while continuously processing queries or calculations for sensor data stream; and on a delivery system with redelivery when data is lost.

Acknowledgments

Part of this research is the result of “Research on High Functional Network Platform Technology for Large-scale Distributed Computing,” which is joint research by NICT and Osaka University. For the evaluation environment, the JGN-X PIAX Testbed (project number JGNX-A12005) was used. We wish to extend our heartfelt gratitude to everyone involved in this research.

References

1 N. Magharei and R. Rejaie, “PRIME: Peer-to-Peer Receiver-Driven Mesh-Based Streaming,” Proc. of IEEE International Conference on Computer Communications (INFOCOM 2007), pp.1415–1423, 2007.

2 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.

3 SAKASHITA Suguru, YOSHIHISA Tomoki, HARA Takahiro, and NISHIO Shojiro, “Methods to Reduce the Number of Peers that Stop Playing Considering the Importance of Divided Data for P2P Streaming Environments,” Information Processing Society of Japan Journal, Vol.52, No.11, pp.3008–3017, 2011.

4 T. A. Le and H. Nguyen, “Application-Aware Cost Function and Its Performance Evaluation over Scalable Video Conferencing Services on Heterogeneous Networks,” Proc. of IEEE Wireless Communications and Networking Conference (WCNC 2012), pp.2185–2190, 2012.

5 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, Vol.22, No.3, pp.1–9, 2014.

6 ISHI Yoshimasa, KAWAKAMI Tomoya, YOSHIHISA Tomoki, and TERANISHI Yuuichi, “Development and Evaluation of A P2P Streaming System for Sensor Data Stream with Different Collection Cycles,” Information Processing Society of Japan Journal, Vol.55, No.2, pp.707–720, 2014.

7 T. Kawakami, Y. Ishi, T. Yoshihisa, and Y. Teranishi, “A Delivery Method considering Communication Loads for Sensor Data Stream with Different Collection Cycles,” Proc. of ACM Symposium On Applied Computing (SAC 2013), pp.611–618, 2013.

8 KAWAKAMI Tomoya, ISHI Yoshimasa, YOSHIHISA Tomoki, and TERANISHI Yuuichi, “An Evaluation of Load Distribution Method on Multi-Source P2P Sensor Data Stream Delivery System,” IEICE Technical Report (Internet Architecture 2013-11), Vol.113, No.94, pp.61–66, 2013.

9 Y. Ishi, T. Kawakami, T. Yoshihisa, and Y. Teranishi, “An Implementation of P2P Sensor Data Streaming System Considering the Number of Hops,” Proc. of IEEE Global Conference on Consumer Electronics (GCCE 2014), pp.428–429, 2014.

10 T. Yoshihisa, Y. Hamaguchi, Y. Ishi, Y. Teranishi, T. Hara, and S. Nishio, “A Sensor Data Aggregation System using Mobile Agents,” Distributed Networks: Intelligence, Security, and Applications, Chapter 2, CRC Press, Taylor and Francis, pp.39–65, 2013.

11 T. Kawakami, Y. Ishi, T. Yoshihisa, and Y. Teranishi, “A Load Distribution Method Based on Distributed Hashing for P2P Sensor Data Stream Delivery System,” Proc. of IEEE Computer Software and Applications Conference (COMPSAC 2014) Workshops, pp.716–721, 2014.

12 I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, “Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications,” IEEE/ACM Transactions on Networking, Vol.11, No.1, pp.17–32, 2003.

13 KAWAKAMI Tomoya, ISHI Yoshimasa, YOSHIHISA Tomoki, and TERANISHI Yuuichi, “A Study of Robustness Enhancement Technique on P2P Sensor Data Stream Delivery System Using Distributed Hashing,” IPSJ SIG Technical Report, Vol.2014-DPS-158, No.3, 6 pages, 2014.

Tomoki YOSHIHISA, Ph.D.

Associate Professor, Cybermedia Center,
Osaka University
Streaming Delivery, Video on Demand

Yuuichi TERANISHI, Ph.D.

Research Manager, New Generation Network
Laboratory, Network Research Headquarters
Ubiquitous Computing, Overlay Network,
Multimedia, Database, Mobile