Journal of Information Processing Vol.20 No.1 154–166 (Jan. 2012)

[DOI: 10.2197/ipsjjip.vol.20.154]

Regular Paper

Adaptive Overlay Network for High-Bandwidth Streaming

Tsuyoshi Hisamatsu[1,a)] Hitoshi Asaeda[1]

Received: April 11, 2011, Accepted: September 12, 2011

Abstract: In this paper, we investigate the limitation caused by reception bandwidth in current overlay streaming, and propose a novel overlay network architecture for high-quality, real-time streaming. The proposed architecture consists of two components: 1) join and retransmission control (JRC), and 2) redundant node selection (RNS). The JRC dynamically adjusts the number of join and data retransmission requests based on the network condition and reception packet fluctuation of the receiver. The RNS selects retransmission nodes based on the retention probability of lost packets requested by the receivers. We have designed and implemented the algorithm of the proposed architecture. According to our evaluation, our approach results in an additional 1–2 Mbps reception bandwidth from existing overlay streaming applications.

Keywords: P2P, overlay network, real-time streaming, multimedia

1. Introduction

Overlay streaming systems such as PPLive [26] have recently become very popular. Overlay streaming does not require any router support for data transmission. Instead, some of the receiver nodes are responsible for forwarding data to other receiver nodes. In overlay streaming applications, the receiver nodes may join or leave the overlay network frequently and randomly. Therefore, different nodes must forward identical data packets (e.g., using MDC [12]) to avoid data loss. However, forwarding a large number of packets consumes peer bandwidth and further degrades the playback quality. The bandwidth limitation of existing overlay streaming applications is less than 1 Mbps, as summarized in **Table 1**. This bandwidth limitation becomes a barrier for the deployment of high-quality overlay streaming services. To support high-quality streaming, this limitation must be overcome by designing a new overlay streaming system that supports a broader transmission path.

In practise, the bandwidth available for overlay streaming is further limited by "weak" receivers that attach using a lower bandwidth or lossy link. **Table 2** lists the last mile bandwidth in the top five countries among countries that deploy high-speed Internet connectivity [3]. According to Akamai's report, the average last mile bandwidth in 125 countries is less than 1 Mbps, whereas the last mile bandwidth in South Korea and Japan, which are the top two countries in terms of last mile bandwidth, is approximately 10 Mbps. The transmitted data quality of overlay streaming depends on the network conditions of the receivers, because most data receivers are tasked with copying streamed data and transmitting it to other receivers in the overlay network. Such a big network bandwidth gap affects the overall streaming quality. Therefore, it is necessary to study how the streaming system appropriately adapts to each receiver's network condition and quickly retransmits lost packets to the lossy receivers, because this is a key factor in designing a new overlay streaming system.

Table 1 Streaming bandwidth of existence research.

Papers	Streaming bandwidth
CoolStreaming [19], [30]	450 Kbps
AnySee [9], [29]	300 Kbps
Chainsaw [27]	600 Kbps/800 Kbps
Paper [21]	480 Kbps/960 Kbps
Paper [25]	1 Mbps
Paper [23]	400 Kbps

Table 2 Top five last mile bandwidths among countries that deploy high-speed Internet.

	Country	Above 5 Mbps	5–10 Mbps	10–15 Mbps	15–20 Mbps	20–25 Mbps	Above 25 Mbps
1	S.Korea	74%	29%	15%	8.6%	5.7%	16%
2	Japan	60%	34%	17%	5.5%	2.0%	1.9%
3	Romania	46%	33%	7.9%	2.4%	1.1%	1.8%
4	Sweden	42%	31%	6.7%	2.2%	0.9%	1.6%
5	HongKong	39%	31%	6.6%	3.9%	2.6%	5.7%

In this paper, we propose a novel overlay real-time streaming architecture. In this study, we focus on high-quality streaming such as MPEG2-TS or H.264 streaming, which warrants a transmission bandwidth of 4,096–16,384 Kbps. This paper is organized as follows. Section 2 describes the background of overlay streaming systems and related work. Section 3 details the problem in overlay streaming by evaluating existing systems. Section 4 describes the technical aspects and the proposed solution for increasing the reception bandwidth. Section 5 describes our approach. Section 6 describes the design and implementation of our approach. Section 7 evaluates the proposed architecture, and Section 8 concludes this paper.

2. Background and Related Work

Many overlay streaming systems employ a tree topology struc-

[1] Graduate School of Media and Governance Keio University, Fujisawa, Kanagawa 252–0882, Japan
[a)] ringo@sfc.wide.ad.jp

ture, which emulates IP multicast (Overcast [13]). To ensure a minimum transmission delay and a bounded node workload (in terms of the fan-out degree), these systems use hierarchical clustering to construct and maintain an efficient structure in a large-scale network. Nevertheless, and its failure often causes buffer underflow errors in a large number of its descendants. To avoid an unbalanced load and to reduce the vulnerability of the tree structure, mTreebone [11] and Bullet [16], [18] adopt a hybrid topology structure that includes a mesh-based tree, and SplitStream [20] adopts a multiple-tree topology structure that maintains multiple distribution trees. Most existing overlay streaming systems focus on the design of the topology structure in trying to address scalability problems.

The requirements of stable overlay streaming systems have been defined by Liu et al. [15] as follows: 1) *Transport bandwidth*: messaging overhead and duplicate packets must be constrained to enhance streaming performance and playback quality. 2) *Start-up delay*: start-up delay should be minimized to avoid inconveniencing users. Moreover, because a playback engine has its own buffer, a two-stage buffer might degrade the performance. Decentralized autonomous control reduces the cost of generating and managing data forwarding. However, there are several ways to discover neighboring nodes in a system [16], [18], [27], [29]. 3) *Transmission delay*: reducing the time taken to find a data-forwarding path helps to minimize the transmission delay. 4) *Fault tolerance*: maintaining redundant forwarding paths and forwarding duplicate packets are necessary.

Magharei et al. [21] compared mesh and tree topology structures, and the mesh topology structure using PRIME [22] was found to be better than the multiple-tree topology structure. A multiple-tree topology structure requires that redundant packets be retransmitted on each sub-tree, for the following reasons: 1) In the multiple-tree topology structure, the forwarding path for each packet is identified using a static mapping between packet descriptions and trees. 2) The forwarding path from a source to individual peer nodes is more stable in the mesh-based approach than in the tree-based approach. 3) The participating peer nodes achieve high bandwidth utilization (≥95%) in the mesh approach, whereas the maximum aggregate bandwidth utilization in the tree approach is only 90%. 4) A deadlock event occurs in the tree-based approach when a tree becomes saturated, and cannot accept a newly joined node.

The tree and multiple-tree topology structures were evaluated by Birrer et al. [24] using GridG [10] and the PlanetLab testbed [6]. They found that the multiple-tree topology structure maximized the data transmission rate and minimized the delay in low-bandwidth/unstable network conditions. The mesh topology structure has a simple structure [15], but large overheads. The tree topology structure has a short start-up time, but has low utilization efficiency for a single transmission path and also involves large overheads.

3. Evaluation of Existing Approaches

The evaluation of existing approaches is the primary input to our design of high-quality overlay streaming.

Table 3 Topology types and bandwidth ranges (Kbps).

A	3,200–11,200	4,000–16,000	4,000–16,000	20,000–40,000
B	800–11,200	1,000–16,000	1,000–16,000	5,000–40,000

3.1 Topology Configuration

We prepared an evaluation network environment using the ModelNet IP emulation framework [4]. Our emulation network consisted of 12 VMware [28] nodes on a MacPro Dual 2.66 GHz Quad-Core Intel Xeon machine, with 2 emulators and 10 simulators. In this emulation, we monitored the condition of each node that was receiving streaming data packets of bandwidth 1–10 Mbps, as described in Akamai's report [3].

Based on the classification described by Calvert et al. [7], we defined network links to be client–stub, stub–stub, transit–stub, and transit–transit, depending on their location in the network, and configured the link bandwidth for each topology, depending on its type. Each link type has an associated bandwidth range uniformly chosen at random. By changing the range, we varied the bandwidth constraints in each topology. For overall simulations, we defined four different topology types corresponding to A and B with typical streaming rates of 1,024–8,192 Kbps, as specified in **Table 3**. Each topology type generated by INET [14] consists of 5,000 edge nodes in the virtual networks, and has 420 users (i.e., overlay participants) for the overlay streaming system. In all simulation instances, the users start receiving streaming data simultaneously.

3.2 Simulation Results

We evaluate three overlay streaming systems – tree (Overcast), hybrid (Bullet), and multiple-tree (SplitStream) topology structures – to compare the distribution of reception bandwidth, and use MACEDON [1] as a common development infrastructure.

In our evaluation, we compare topology types A and B, as a big network bandwidth gap exists between the topology of each, and hence they are a good example of the impact on the reception bandwidth of the receiver. **Figure 1** shows the evaluation results in topology type A. The vertical error bars show the minimum and maximum values. In topology type A, the reception bandwidth in the hybrid topology is higher than in the multiple-tree topology in all sessions, because the retransmission of the lost packets (which is necessary to achieve high-quality overlay streaming) is triggered by the mesh topology structure. **Figure 2** shows the evaluation results in topology type B. In topology type B, the reception bandwidth in both hybrid and multiple-tree topology structures was only about 4,500 Kbps. Based on the simulation results, the network bandwidth gap causes an overall low reception bandwidth. The hybrid topology structure causes packet loss in the start-up phase due to the bottleneck links, and packet retransmission decreases the reception bandwidth. In the multiple-tree topology structure, the reception bandwidth is quite steady across all the sessions.

According to the simulation results, to design a high-quality overlay streaming system, it is necessary to focus not only on the scalability and stability of the topology, but also on the effective utilization of the receiver network bandwidth.

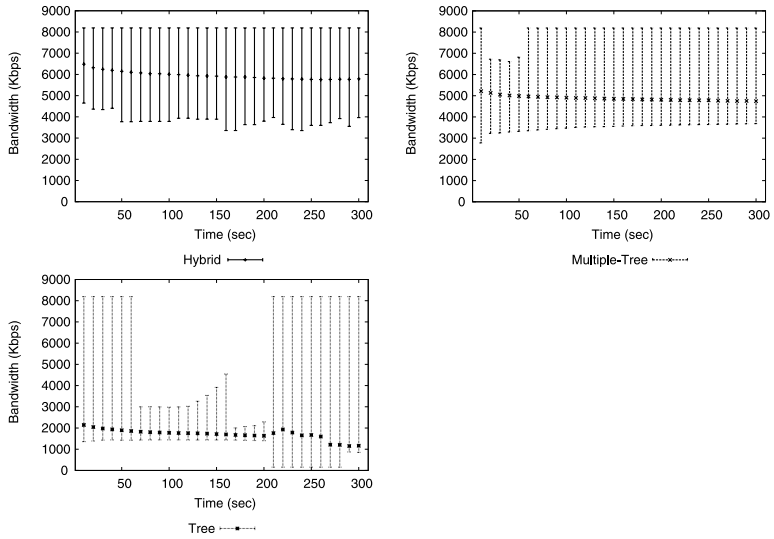

Fig. 1 Distribution of reception bandwidth. The vertical error bars show the minimum and maximum values (Topology type A, 8,192 Kbps).

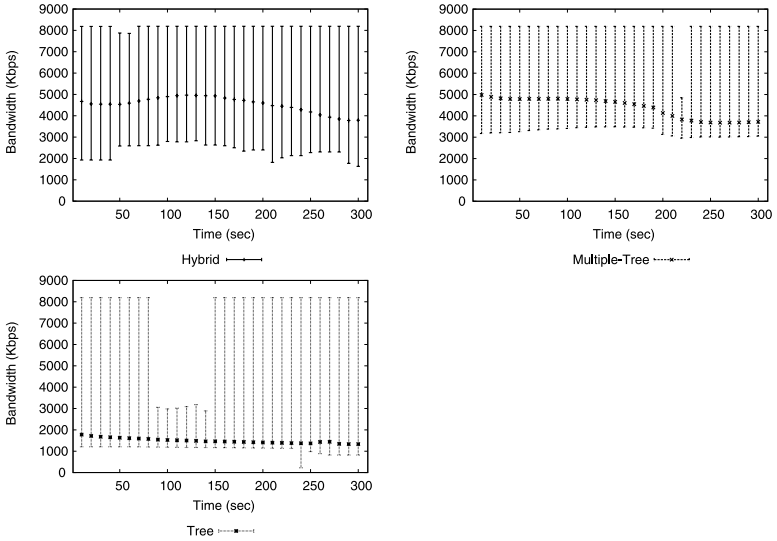

Fig. 2 Distribution of reception bandwidth. The vertical error bars show the minimum and maximum values (Topology type B, 8,192 Kbps).

4. Technical Aspects

In this section, we describe the technical aspects of efficiently enhancing the reception bandwidth.

4.1 Number of Sessions

Sachin et al. [23] evaluated the tree and mesh topology structures on the Internet with a transmission bandwidth of 400 Kbps. The mesh topology structure was found to have many duplicated data and control messages, and led to low reception bandwidth.

We compare the effect of attributes on join requests to multiple-tree sessions and retransmission requests that handle lost packets. We examine Bullet, which adopts a hybrid topology structure and fixes the number of sessions. **Figure 3** shows the simulation results with various join requests to multiple-tree sessions, which is denoted as J hereafter. The default number of join requests to multiple-tree sessions is fixed at 10 for Bullet. Here, J can take values of 5, 10, 20, and 30. In topology type B, which has a high network bandwidth gap at the receiver and uncertain usable bandwidth, $J = 20$, which has high redundancy, achieves over 6 Mbps in total. A value of $J = 5$ has between 500 Kbps –and 1 Mbps lower reception bandwidth than $J = 20$. A value of $J = 10$ has lower reception bandwidth than both $J = 5$ and $J = 20$. $J = 10$ has better reception bandwidth than $J = 30$ with join requests to multiple-tree sessions, and if it is more than 20 sessions, it consumes the usable bandwidth and breaks the redundancy.

Figure 4 shows the simulation results with various retransmission requests, which we denote as R from here on. The default number of retransmission requests is fixed as 10 for Bullet. Here, R can take values of 5, 10, 20, and 30. In topology type B, $R = 30$ achieves high reception bandwidth during the start-up phase, but the reception bandwidth becomes unstable as the streaming proceeds. A value of $R = 10$ or $R = 20$ has a lower messaging cost than $R = 30$, but the reception bandwidth reduces to about 500 Kbps as the streaming proceeds, with low redundancy. Val-

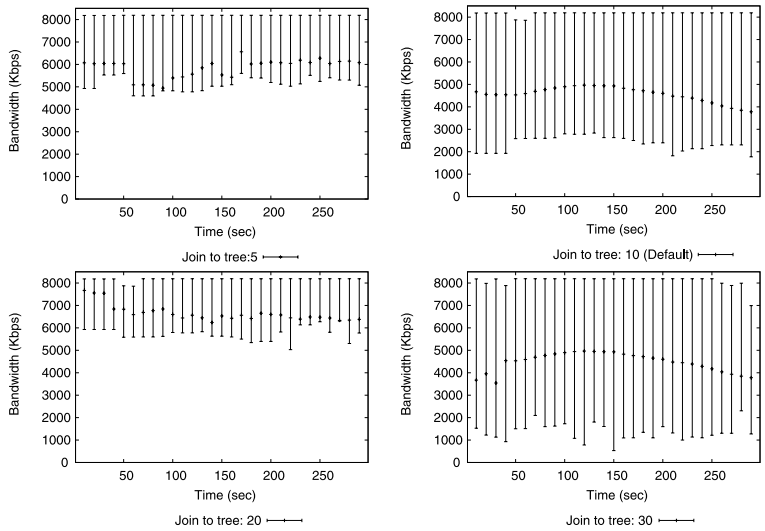

Fig. 3 Distribution of reception bandwidth – changing the number of join requests to multiple-tree sessions. The vertical error bars show the minimum and maximum values (Topology type B, 8,192 Kbps).

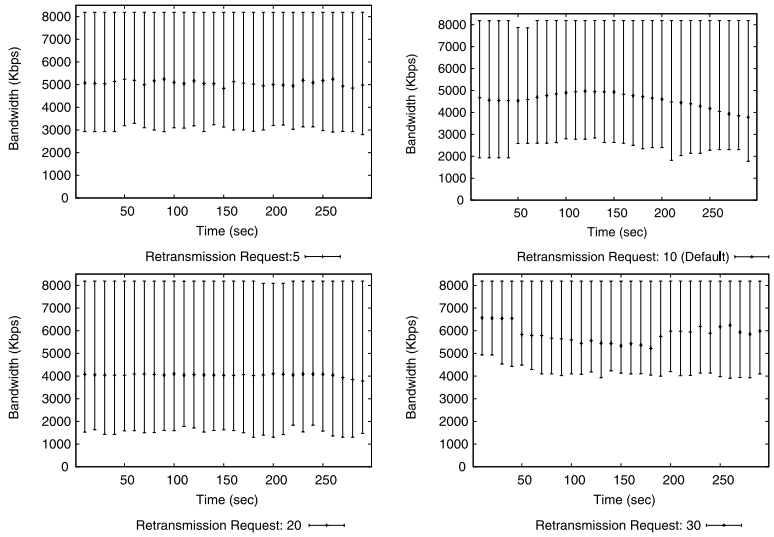

Fig. 4 Distribution of reception bandwidth – changing the number of retransmission requests. The vertical error bars show the minimum and maximum values (Topology type B, 8,192 Kbps).

ues of $R = 20$ and $R = 30$ complement the lost packets, but the reception bandwidth reduces with high messaging cost. In the high network bandwidth gap at the receiver, a low number of retransmission requests results in high reception bandwidth, but it has the limitation of complementing lost packets. Therefore, there is a tradeoff between effectively utilizing network bandwidth and the stability of the reception bandwidth. To keep the messaging cost low, it is better to control the number of sessions and set it to a low number whenever the network bandwidth gap is small.

After comparing the changing conditions caused by join requests to the multiple-tree and retransmission requests, we observed the following: 1) The number of sessions must be flexible, because they are affected by the network bandwidth of the receiver. 2) The number of join requests to multiple-tree sessions determines the redundancy, and potentially increases the reception bandwidth. 3) The messaging cost of retransmission requests is higher than that of join requests to multiple-tree sessions, and lowers the reception bandwidth.

4.2 Node Selection

We focus now on the retention probability of lost packets for redundant node selection. **Figure 5** shows the redundant node connection based on the common tree topology structure. In the tree topology structure, when an upper node loses packets, its child nodes lose the same packets too. In the case of redundant nodes connected in a neighborhood, there is a high possibility that a destination node will connect to the same parent node. Because node B is in the same neighborhood as node A, when node A requests that node B retransmits lost packets, it is highly possible that node B does not have the requested packets. Bullet randomly connects the redundant nodes using RanSub [17]. Random node selection might have a higher probability of obtaining the requested packets from the randomly connected node than from the neighborhood. However, random node selection might have a high delay in retransmitting lost packets due to the long round trip time (RTT) between the nodes.

Here, we show the following technical aspects: 1) Effective re-

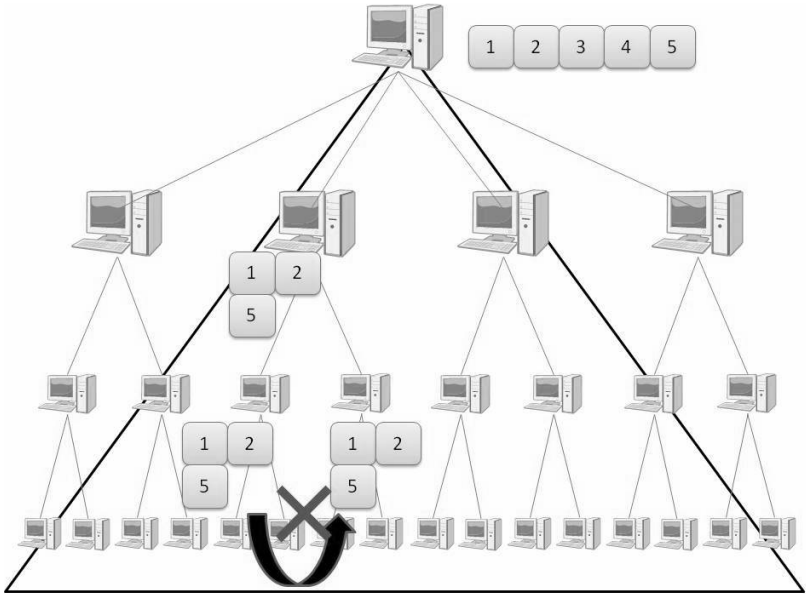

Fig. 5 Redundant nodes based on previous Tree connection.

dundant node selection must consider the retention probability of lost packets and depend on the tree topology structure in order to enhance the reception bandwidth. 2) Adopting node selection must consider the RTT as a parameter in deciding on nodes.

5. Approach

In this section, we describe our approach on achieving high-quality overlay streaming based on several considerations in Sections 3 and 4. **Figure 6** shows overview of this system. This system consists of the combination of multiple-tree and retransmission of the lost packets. We propose JRC, which controls the value of sessions to adapt to each receiver's network environment, and RNS, which is monitoring function for overlay network, to trigger the JRC.

5.1 Join and Retransmission Control (JRC)

According to the analysis in Section 4.1, the number of sessions decides the impact on usable bandwidth consumption and the retransmission condition of lost packets. We define the "join and retransmission control (JRC) component" in the proposed architecture by modifying the approaches that exist in join requests to multiple-tree sessions and retransmission requests. As the parameter to trigger a change is the number of sessions, we incorporate "Reception Packet Fluctuation (RPF) monitoring" at each receiver.

In general, unicast and multicast real-time streaming systems such as DVTS [5] adopt congestion control mechanisms based on the packet loss rate. The congestion control function is triggered by packet loss. However, this technique is not applicable for overlay streaming. It is difficult to detect the packet loss in overlay streaming systems, because overlay streaming transmits identical data packets encoded by, for example, MDC [12], and usually only the streaming applications recognize the packet loss after the data is assembled.

RPF monitoring can be used to detect packet loss by comparing the received packets within the monitoring period. **Figure 7**

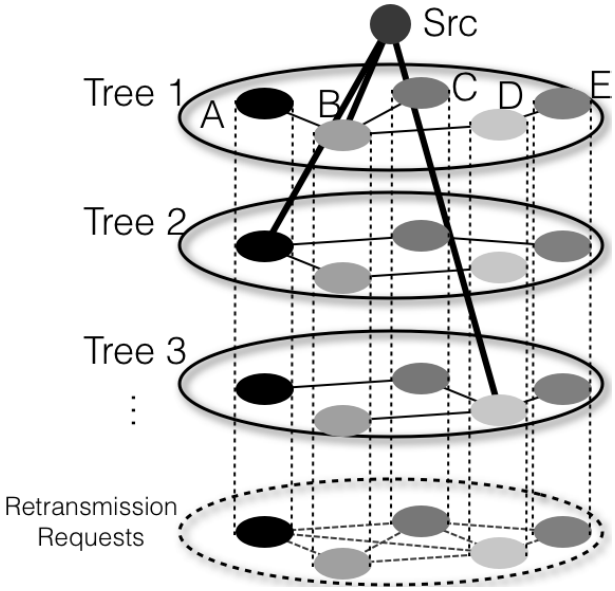

Fig. 6 Overview of this system.

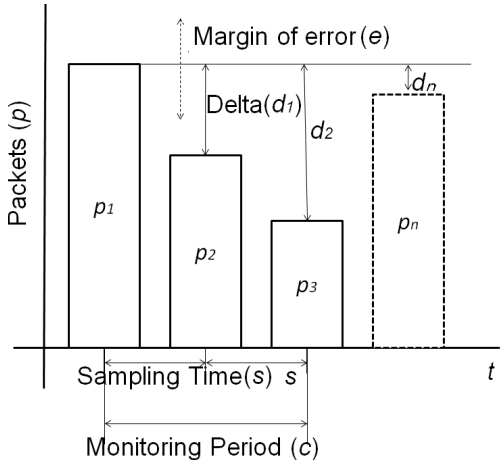

Fig. 7 Overview of RPF.

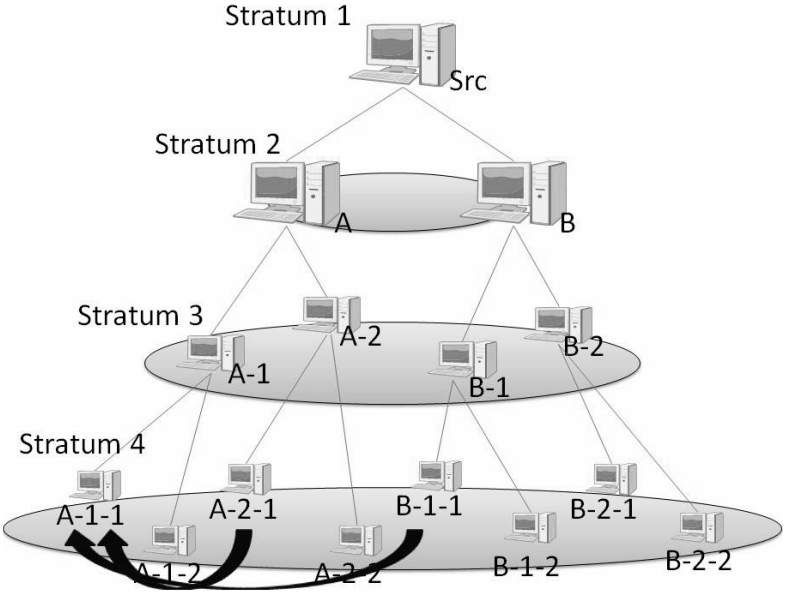

Fig. 8 Redundant node selection based on the distance of relationship.

shows an overview of RPF. A data receiver counts the received packets, p, per sampling time unit, s, and calculates the differences, d, in the number of packets in the period, c. We define the margin of error, e, for deciding on the packet loss event. As shown in Fig. 7, if the observed d_n is more than e in the monitoring period c, RPF decides a packet loss has occurred, and the JRC procedure is triggered.

5.2 Redundant Node Selection (RNS)

According to the discussion in Section 4.2, we define the "redundant node selection (RNS) mechanism" that selects retransmission nodes based on the retention probability of lost packets requested by receivers. **Figure 8** illustrates the RNS. The RNS mechanism uses the following parameters for redundant node selection:

- RTT
- Distance of relationship
- Stratum

To minimize the transmission delay in the overlay streaming system, the mechanism must consider the RTT as one of the parameters for redundant node selection.

The "distance of relationship" is another parameter that affects the path of a packet between a node that requests lost packets, and a "potential node" that will give the lost packets to the requesting node. For instance, as seen in Fig. 8, the distance from node A-1-1 to node B-2-2 is further than that from node A-1-1 to node A-2-2. If node A-1-1 loses packets, node B-2-2, which is further away, is selected as the potential node, because it has a higher capability of retaining lost packets.

If all nodes connect to the source or the same parent nodes, the whole overlay streaming network will be unstable [11]. We consider the stability of the overlay network by verifying the depth of the tree structure. We define the "Stratum" number to categorize the nodes that are at the same depth of the tree structure. As seen in Fig. 8, nodes A1, A2, B1, and B2 are at Stratum 3. To ensure the stability of overlay streaming in the tree structure, node A1 should connect to nodes in the same Stratum number, or lower.

6. Design and Implementation

In this section, we describe the design of JRC based on Section 5.1 and redundant node selection based on Section 5.2.

6.1 JRC

The Join and Retransmission Control (JRC) component is implemented with the Reception Packet Fluctuation (RPF) monitoring function as described in Section 5.2. The processing flow of JRC and the RPF function are shown in **Fig. 9**. In this flow, if packet loss is continuously observed during the RPF monitoring period, RPF changes the state to negative. If it is negative, JRC proceeds with the following three phases to improve the reception bandwidth. 1) When RPF observes packet loss during its monitoring period, JRC assumes reception bandwidth shortage and the number of retransmission requests rns is increased ($PHASE\_1$), 2) If RPF observes packet loss continuously occurring, JRC supposes the network bandwidth shortage and slowly reduces the number of join requests to multiple-tree sessions $trees$ to MIN_TREES ($PHASE\_2$), and 3) If packet loss is additionally observed by RPF, JRC supposes that its own traffic might choke its own usable bandwidth and slowly reduces the number of retransmission requests rns to MIN_RNS ($PHASE\_3$). However, if no packet loss is observed, both the number of join requests to multiple-tree sessions $trees$ and the number of retransmission requests rns are slowly increased to extend reception bandwidth to MAX_TREES and MAN_RNS.

6.2 RNS

Each node is assigned a unique ID by the parent node. When a node joins the overlay network, its parent node generates a fixed digit number for the node. The node ID is appended to the node along with the parent ID and the digit number, and saved by the node. As we use a multiple-tree topology structure, node IDs are dependent on each tree and managed by a bit operation.

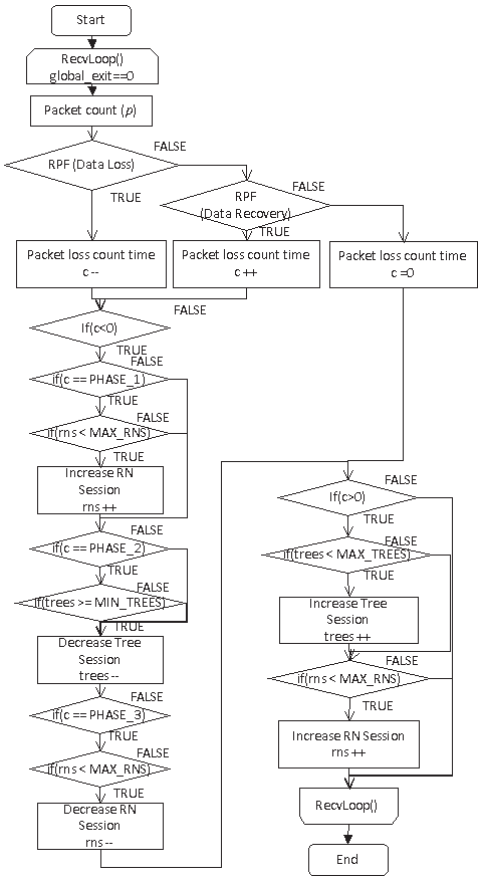

Fig. 9 JRC flow.

We calculate the distance of relationship using the track record of the parent nodes. To diversify the referential node, we use RanSub to discover potential nodes. Discovered nodes are detected as redundant nodes and set priorities, based on the discussion in Section 5.2. We use the RTT between nodes to compare neighborhoods. The RTT is considered at the beginning because it is not necessary to connect a node that has requested packets, and has sufficient performance to retransmit the packets, but is too far away. Calculating the RTT from each discovered node tells us how the network is dynamically changing. To compare the distance of relationship, we use a structure of node IDs from the parents linked by fixed digits. The node ID is split by fixed digits and then compared to the upper level. This determines the branch stratum level and decides the priority of a retransmission request. From this node ID assignment method, the shorter a node ID length, the higher the stratum of the node, and vice versa. Therefore, we refer to the node ID length to determine the stratum. A node connects to the redundant node that has the longest node ID, because it is recommended that it connects to a redundant node in the same or lower stratum, as discussed in Section 5.2. If packet loss occurs, each node passes a request down a generated redundant node list.

7. Evaluations

7.1 Parameters

Table 4 shows the parameters and values we obtained in our evaluation. All the evaluation scenarios and results are explained in the following sections. $DEFAULT\_TREES$ and $DEFAULT\_RNS$ are the start-up values of join requests to multiple-tree sessions and retransmission requests based on the default values of Bullet and SplitStream. MIN_TREES is set at half the value of $DEFAULT\_TREES$. As we described in Section 4.1, the value of the sessions achieving highest reception bandwidth is 20, but is less unstable than 10. We set MAX_TREES to the value intermediate between 20, which achieved highest average reception bandwidth, and 10, which achieved the most stable reception bandwidth. From Section 4.1, too many retransmission requests set up unstable reception bandwidth. Additionally, as we described in Section 4.2, the retransmission requests are essential to achieve high reception bandwidth, and do not require high bandwidth per 1 session like join requests to multiple-tree sessions. Therefore, we put the value of MIN_RNS on 0.8 times the $DEFAULT\ RNS$, and put the value of MAX_RNS on 1.5 times the $DEFAULT\_RNS$.

Table 4 Parameter settings of JRC.

Parameter	Value
MAX_TREES	15
MIN_TREES	5
DEFAULT_TREES	10
MAX_RNS	15
MIN_RNS	8
DEFAULT_RNS	10
PHASE_1_CNT	5
PHASE_2_CNT	10
PHASE_3_CNT	15

7.2 JRC

7.2.1 Number of Sessions

We compared the distribution of reception bandwidth between a system using JRC and a fixed number of sessions. We evaluated four different parameters: 1) JRC (dynamic sessions), 2) MAX (number of join requests to a multiple-tree: 15 sessions, retransmission requests: 15 sessions), 3) MID (number of join requests to a multiple-tree: 10 sessions, retransmission requests: 10 sessions), 4) MIN (number of join requests to a multiple-tree: 5 sessions, retransmission requests: 8 sessions). The simulation results in **Fig. 10** show that the system using JRC has better reception bandwidth in all sessions, with MID as the second best option. MAX and MIN have a similar distribution of reception bandwidth. The distribution of the reception bandwidth of MIN decreased because the number of transmitted packets was too few, while the distribution reception bandwidth of MAX decreased because too many sessions put pressure on the usable bandwidth. Therefore, too many or too few sessions lead to a lower reception bandwidth. The JRC component in our proposed architecture proved to be effective in enhancing the reception bandwidth.

7.2.2 Order of JRC Parameters Priority

Our algorithm, shown in Fig. 9, defined three phases for controlling the number of sessions. In this section, we evaluate three parameters, namely the preferentially deleted join request to a multiple-tree, the preferentially added join request to a multiple-tree, and JRC. These parameters are based on the order of the priority of the redundant nodes with JRC settings of $rns++$, $trees--$, and $rns--$. The preferentially added join request to a multiple-tree setting is $trees++$, $rns--$, and $trees--$,

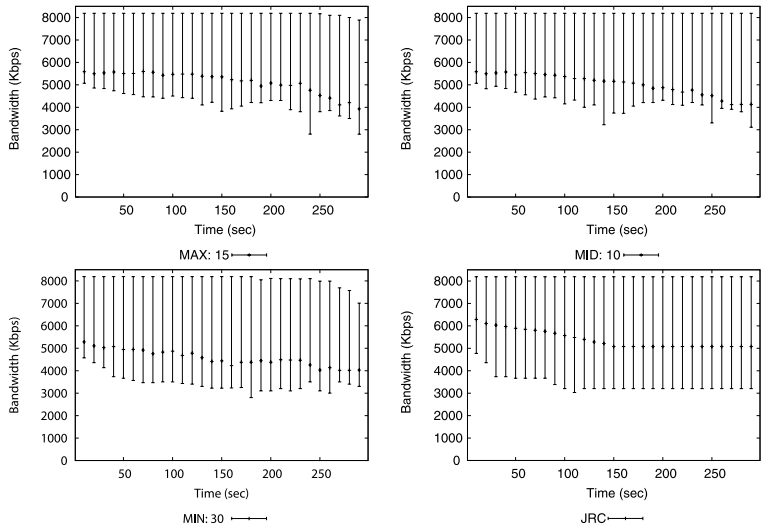

Fig. 10 Distribution of reception bandwidth – JRC. The vertical error bars show the minimum and maximum values (Topology type B, 8,192 Kbps).

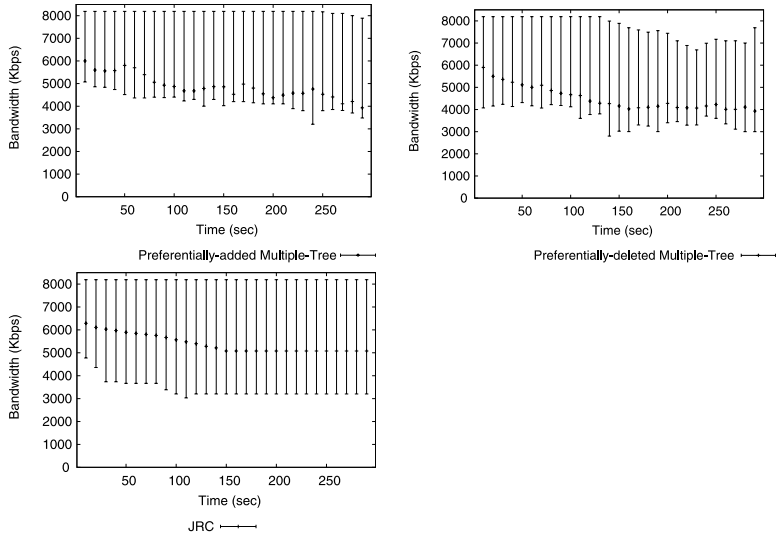

Fig. 11 Distribution of reception bandwidth – order of JRC Parameters. The vertical error bars show the minimum and maximum values (Topology type B, 8,192 Kbps).

and the preferentially deleted join request to a multiple-tree setting is $trees--$, $rns++$, and $rns--$. **Figure 11** shows that the JRC setting has a better distribution of the reception bandwidth than the other two. The preferentially added multiple-tree setting has a higher reception bandwidth than the preferentially deleted multiple-tree setting, because there are more incoming packets from the join request to the multiple-tree in one session than from the redundant node. Therefore, increasing and decreasing the number of join requests to multiple-tree sessions will overrun or underrun the usable bandwidth.

7.2.3 Margins of Error

Packet loss occurrence is monitored by using the RPF to trigger the JRC. As we described in Section 5.1, we considered margins of error, and compared values of 0.3 (30%, JRC setting), 0.1 (10%), and 0.5 (50%). **Figure 12** shows the evaluation results, in which the 0.3 and 0.1 margins of error have similar reception bandwidth at start-up, but the 0.3 margin of error has higher reception bandwidth in overall sessions after 100 seconds. The 0.1 margin of error has too many JRC processes, because packet loss occurred frequently. The reception bandwidth of the 0.5 margin of error is lower than the others because packet loss rarely occurred, which triggered few JRC processes. According to this evaluation, JRC based on RPF monitoring successfully enhances the reception bandwidth.

7.2.4 Sampling Time

We observed the transition of reception packets per sampling time for the RPF calculation in the system. We compared the distribution of reception bandwidth by changing the sampling times. **Figure 13** shows the results of the verification. We evaluated 1 second (JRC setting), 2 seconds, and 5 seconds. The setting of 1 second achieved the broadest reception bandwidth in all sessions. The setting of 2 seconds achieved a broader bandwidth than did the 5 second setting at start-up. However, the bandwidth decreased drastically after 50 seconds, and balanced out at about 3.5 Mbps after approximately 250 seconds.

Based on this evaluation, we concluded that the sampling time is an important factor when calculating the RPF. In addition, a shorter sampling time had a higher availability in the evaluation

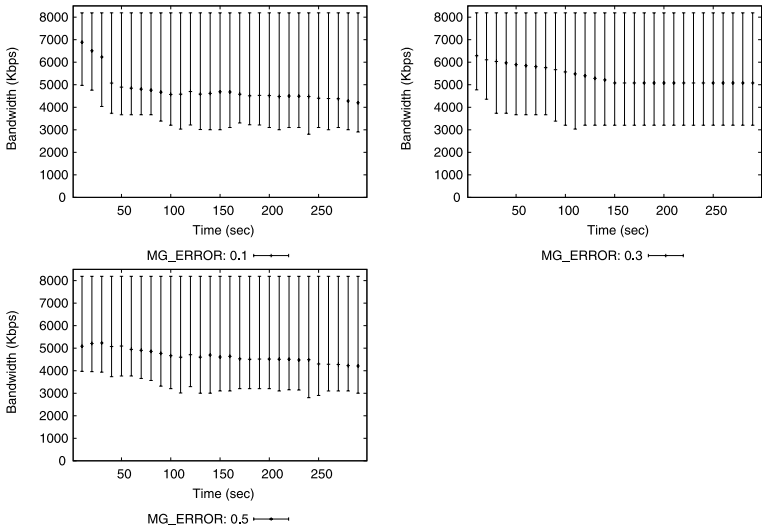

Fig. 12 Distribution of reception bandwidth – margins of error. The vertical error bars show the minimum and maximum values (Topology type B, 8,192 Kbps).

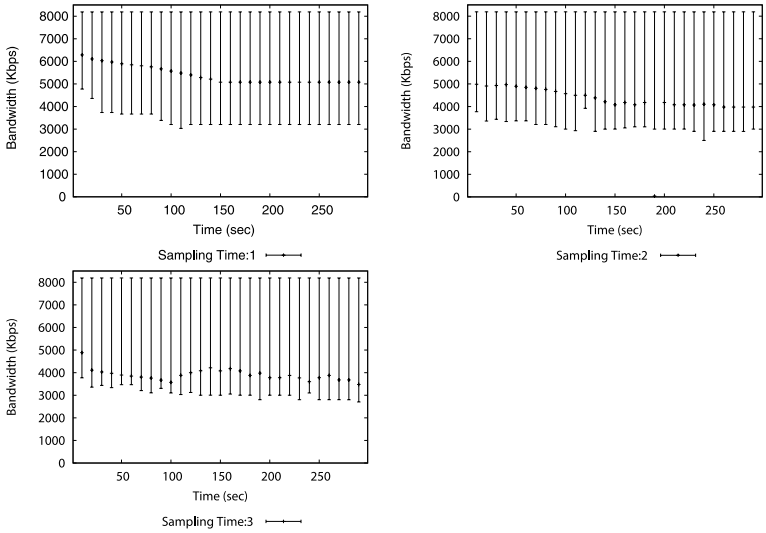

Fig. 13 Distribution of reception bandwidth – sampling time. The vertical error bars show the minimum and maximum values (Topology type B, 8,192 Kbps).

environment.

7.3 RNS

We evaluated the availability of the distance of the relationship for redundant node selection. Our approach uses "RTT priority node selection" described in Section 5.2. We compared the reception bandwidth between RTT-priority node selection and the following functions: 1) "RTT-based node selection" selects the nodes in the order corresponding to short RTT. 2) "distance of relationship-based node selection" selects the nodes in the order corresnponding to long distance of relationship. 3) "stratum-based node selection" selects the nodes which join to same or lower stratum in the tree topology structure. 4) "Random node selection" selects the nodes randomly.

The RTT-based approach achieved broader reception bandwidth within some receivers, but gradually decreased (**Fig. 14**). It is inefficient to retransmit lost packets from the neighborhood under conditions of a high bandwidth network gap. The distance of relationship-based approach achieved broader reception bandwidth in the start-up phase, but also gradually decreased, and was narrower than the RTT-priority approach after 250 seconds. Redundant node selection needs to consider RTT as one of the parameters, because it needs to consider the size of the buffer when supporting real-time streaming. The random node selection is not sufficient to retransmit lost packets in topology type B. Redundant node selection based on specific parameters is needed to enhance retransmission efficiency. Based on these evaluations, we show the usability of redundant node selection based on multiple parameters: RTT, distance of relationship, and stratum.

We compared the distribution of reception bandwidth by changing the order of parameters: 1) "RTT priority" examines RTT first, then distance of relationship and stratum. 2) "distance of relationship priority" examines distance of relationship first, then RTT and stratum. 3) "stratum priority" examines stratum at first, then RTT and distance of relationship.

The stratum priority was narrowest in topology type B, which has a high network bandwidth gap (**Fig. 15**). At start-up, the distance of relationship priority achieved the broadest reception

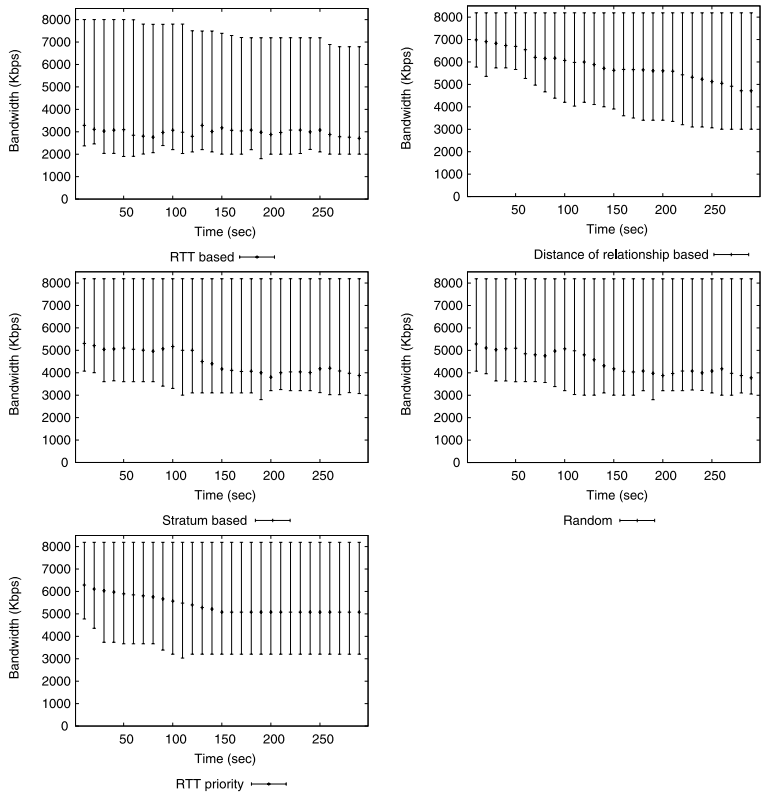

Fig. 14 Distribution of reception bandwidth – redundant node selection. The vertical error bars show the minimum and maximum values (Topology type B, 8,192 Kbps).

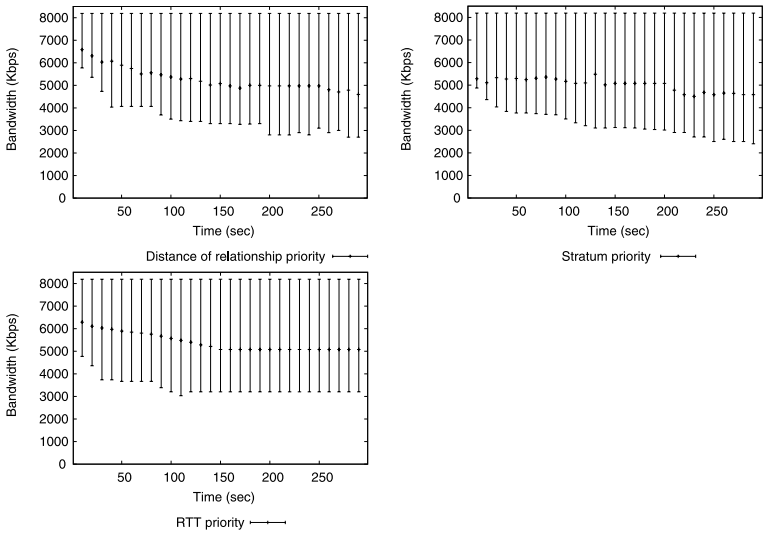

Fig. 15 Distribution of reception bandwidth – redundant node selection. The vertical error bars show the minimum and maximum values (Topology type B (below), 8,192 Kbps).

bandwidth, but gradually decreased. As a result, the RTT priority comprehensively achieved broader reception bandwidth. Based on these evaluations, we preferred using RTT for node selection as an effective way to enhance reception bandwidth.

7.4 Other Existing Overlay Streaming System

We compared the distribution of reception bandwidth between this system and three existing systems in Section 3 to show the effectiveness of this system. **Figure 16** shows the evaluation results in topology type B. In the case of transmission bandwidth 8,192 Kbps, the reception bandwidth of this system and the hybrid system are similar before about 150 seconds, because the JRC is adapting to the network bandwidth of the receiver. After 270 seconds, this system achieved approximately 1 Mbps higher reception bandwidth than the hybrid and multiple-tree systems. Based on this evaluation, this system is verified under a high network bandwidth gap.

We make three observations in these evaluations: 1) Too many or too few sessions lead to reduced reception bandwidth. 2) Controlling the number of sessions leads to enhanced reception bandwidth. 3) Monitoring RPF and setting adequate sampling times and margins of error lead to enhanced reception bandwidth. We

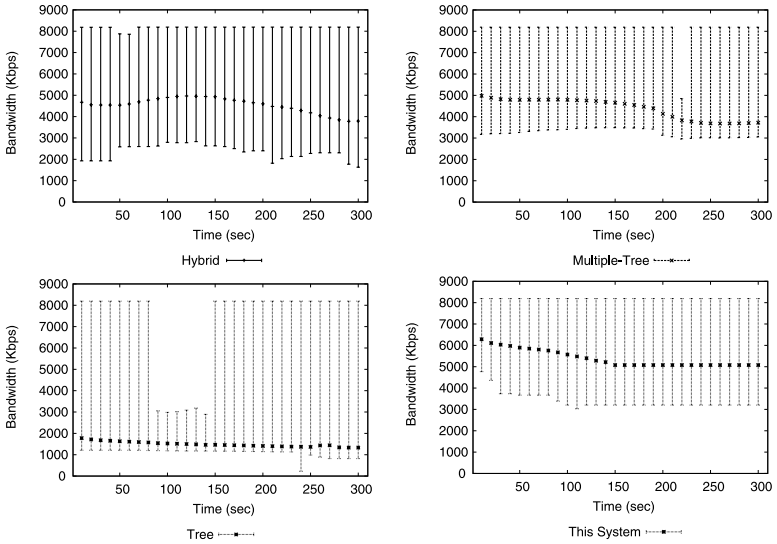

Fig. 16 Distribution of Reception Bandwidth. The vertical error bars show the minimum and maximum values (Topology type B, 8,192 Kbps).

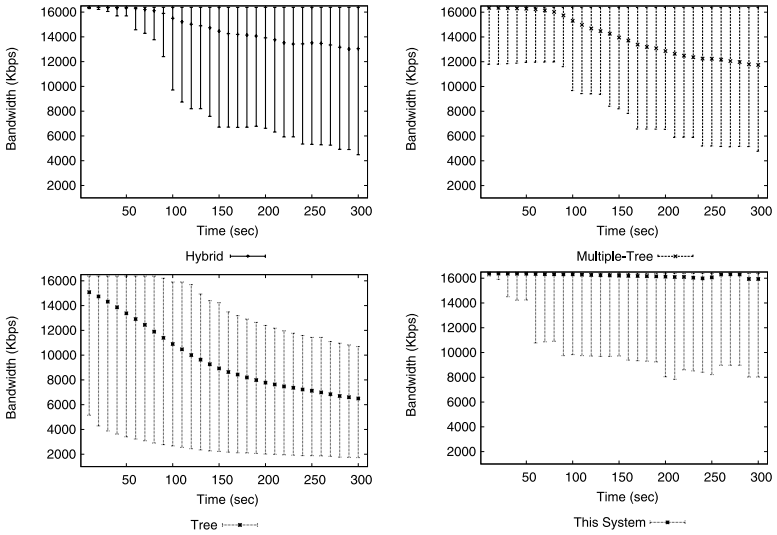

Fig. 17 Distribution of Reception Bandwidth on PlanetLab. The vertical error bars show the minimum and maximum values (16,384 Kbps).

validated the RPF as a trigger for JRC based on aspects of the overlay network.

7.5 PlanetLab: Other Existing Overlay Streaming Systems

In order to compare our system and existing overlay streaming systems in the real-world, where there exist many routers, complicated network topologies, and uncertain traffic, we conducted our experiments on PlanetLab. Our experiments involved almost all the active nodes of PlanetLab, with the total number ranging from 200 to 300 during our experiment period (February, 2011). We evaluated three overlay multicast systems with different topologies, namely– tree, hybrid, and multiple-tree topologies – using the MACEDON common development infrastructure. Each active PlanetLab node runs a copy of MACEDON. The bootstrap node is located in U.S.A. (pl2.cs.yale.edu). We evaluated with a transmission bandwidth of 16,384 Kbps, because most PlanetLab nodes connect over 1 Gbps and do not have network bandwidth limitation in contrast to our evaluation environment (Section 3.1)

Figure 17 shows the evaluation result using MACEDON on PlanetLab. The reception bandwidth of this system, the hybrid system, and the multiple-tree system are similar before about 50 seconds. After 50 seconds, the hybrid system and the multiple-tree system reduced to about 3–4 Mbps. Our system was stable and continued to achieve over 16 Mbps. In addition, the worst reception bandwidth receiver in this system was better than the others, at about 2 Mbps.

We make two observations from these evaluations: 1) JRC and Monitoring RPF as a trigger for JRC enhance all receivers' reception bandwidth in the real-world. 2) Overlay streaming which adapts to the receiver's network condition is able to achieve a higher reception bandwidth using multiple parameters: RTT, distance of relationship, and stratum.

This research was performed using MDC not Network Coding [2] for the following reasons: 1) In a real-life overlay streaming system, receivers constantly join and leave an overlay net-

work. This means the overlay structure is changing constantly, and it would be very hard for the network to continuously adapt perfectly to the overlay structure [8]. 2) This approach, using JRC, could not decide on a unique route and could not make full use of the characteristics of network coding.

8. Conclusions

In this paper, we proposed a novel overlay network architecture for high-quality, real-time streaming focused on the network conditions and bandwidth gaps at the receiver. We observed the reception bandwidth decreases caused by the network bandwidth gap at the receiver in simulations, and showed that controlling the number of sessions is an effective technique to enhance the reception bandwidth. We proposed an architecture consisting of: 1) JRC to adjust the number of join and data retransmission requests, based on the receivers' network condition and RPF, 2) RNS to select retransmission nodes based on the retention probability of lost packets requested by receivers. We designed and implemented an overlay streaming system based on the proposed approach, and evaluated it. According to the evaluation, we verified the behaviors of both JRC and RNS. Based on the comparisons, this system achieves an additional 1–2 Mbps reception bandwidth above existing overlay streaming systems, and is verified under a high network bandwidth gap and in the real-world. The results of this study reduces the bandwidth limitation of overlay streaming, and demonstrates a novel architecture for the high-quality streaming services in the future.

Reference

[1] Adolfo, R., Charles, K., Sooraj, B., Dejan, K. and Amin, V.: MACEDON: Methodology for automatically creating, evaluating, and designing overlay networks, *NSDI'04*, Berkeley, CA, USA, USENIX Association, pp.267–280 (2004).

[2] Ahlswede, R., Cai, N., Li, S. and Yeung, R.: Network information flow, *IEEE Trans. Information Theory*, Vol.46, No.4, pp.1204–1216 (2000).

[3] Akamai Technologies, Inc.: 2nd Quarter, 2009 The State of the Internet (2010), available from ⟨http://www.akamai.com/dl/whitepapers/Akamai_State_Internet_Q3_2009.pdf⟩.

[4] Amin, V., Ken, Y., Kevin, W., Priya, M., Dejan, K., Jeff, C. and David, B.: Scalability and Accuracy in a Large-Scale Network Emulator, *OSDI*, pp.271–284, ACM, New York, NW, USA (2002).

[5] Ogawa, A., Kobayashi, K., Sugiura, K., Nakamura, O. and Murai, J.: Design and Implementation of DV based video over RTP, *Packet Video 2000* (2000).

[6] Brent, C., David, C., Timothy, R., Andy, B., Larry, P., Mike, W. and Mic, B.: PlanetLab: An overlay testbed for broad-coverage services, *SIGCOMM Comput. Commun. Rev.*, Vol.33, No.3, pp.3–12 (online), DOI: 10.1145/956993.956995 (2003).

[7] Calvert, K., Doar, M., Nexion, A. and Zegura, E.: *Modeling Internet Topology* (1997).

[8] Chiu, D.M., Yeung, R.W., Huang, J. and Fan, B.: Can network coding help in p2p networks, *In Second Workshop of Network Coding, in conjunction with WiOpt* (2006).

[9] Zhang, C., Jin, H., Deng, D., Yang, S., Yuan, Q. and Yin, Z.: Anysee: Multicast-based Peer-to-Peer Media Streaming Service System, *IEEE APCC*, pp.274–278 (2005).

[10] Lu, D. and Dinda, P.: GridG: Generating realistic computational grids, *ACM Sigmetrics Performance Evaluation Review*, Vol.30, No.4, pp.33–40 (2003).

[11] Wang, F., Xiong, Y. and Liu, J.: mTreebone: A Hybrid Tree/Mesh Overlay for Application-Layer Live Video Multicast, *Proc. IEEE ICDCS 2007*, p.49 (2007).

[12] Goyal, V.K.: Multiple description coding: Compression meets the network, *IEEE Signal Processing Magazine*, Vol.18, No.5, pp.74–93 (2001).

[13] Jannotti, J., Gifford, D.K., Johnson, K.L., Kaashoek, F.M. and O'Toole, J.W.: Overcast: Reliable multicasting with on overlay network, *OSDI'00*, Berkeley, CA, USA, USENIX Association, pp.197–212 (2000).

[14] Jin, C., Chen, Q. and Jamin, S.: Inet: Internet Topology Generator, Technical Report CSE-TR-433-00, University of Michigan at Ann Arbor (2000).

[15] Liu, J., Rao, S.G., Li, B. and Zhang, H.: Opportunities and Challenges of Peer-to-Peer Internet Video Broadcast, *Proc. IEEE*, Vol.96, No.1, pp.11–24 (2008).

[16] Kostic, D., Braud, R., Killian, C., Vandekieft, E., Anderson, J.W., Snoeren, A.C. and Vahdat, A.: Maintaining High Bandwidth under Dynamic Network Conditions, *USENIX ATC*, USENIX Association, pp.193–208 (2005).

[17] Kostic, D., Rodriguez, A., Albrecht, J.R., Bhirud, A. and Vahdat, A.: Using Random Subsets to Build Scalable Network Services, *USITS*, p.19 (2003).

[18] Kostić, D., Snoeren, A.C., Vahdat, A., Braud, R., Killian, C., Anderson, J.W., Albrecht, J., Rodriguez, A. and Vandekieft, E.: High-Bandwidth Data Dissemination for Large-Scale Distributed Systems (2008).

[19] Li, B., Keung, G.Y., Lin, C., Liu, J. and Zhang, X.: Inside the New Coolstreaming: Principles, Measurements and Performance Implications, *INFOCOM 2008*, pp.1031–1039 (2008).

[20] Castro, M., Druschel, P., Kermarrec, A., Nandi, A., Rowstron, A. and Singh, A.: SplitStream: High-bandwidth content distribution in cooperative environments, *IEEE IPTPS 2003*, pp.298–313 (2003).

[21] Magharei, N., Rejaie, R. and Guo, Y.: Mesh or Multiple-Tree: A Comparative Study of Live P2P Streaming Approaches, *IEEE INFOCOM 2007*, pp.1424–1432 (2007).

[22] Rejaieand, R. and Magharei, M.: PRIME: Peer-to-Peer Receiver-driven Mesh-based Streaming, *IEEE INFOCOM 2007*, pp.1415–1423 (2007).

[23] Sachin, A., Pal, S.J., Aditya, M., Pierpaolo, B. and Bernd, G.: Performance of P2P live video streaming systems on a controlled test-bed, *TridentCom'08*, pp.1–10, ICST, Brussels, Belgium (2008).

[24] Birrer, S. and Bustamante, F.E.: A Comparison of Resilient Overlay Multicast Approaches, *IEEE Journal on Selected Areas in Communications*, Vol.25, No.9, pp.1695–1705 (2007).

[25] Seibert, J., Zage, D., Fahmy, S. and Nita-Rotaru, C.: Experimental Comparison of Peer-to-Peer Streaming Overlays: An Application Perspective, pp.20–27, IEEE Computer Society, *IEEE LCN*, Washington, DC, USA (2008).

[26] Spoto, S., Gaeta, R., Grangetto, M. and Sereno, M.: Analysis of PPLive through active and passive measurements, *IPDPS'09: Proc. 2009 IEEE International Symposium on Parallel&Distributed Processing*, pp.1–7, IEEE Computer Society, Washington, DC, USA (2009).

[27] Pai, V.S., Kumar, K., Tamilmani, K., Sambamurthy, V. and Mohr, A.E.: Chainsaw: Eliminating Trees from Overlay Multicast, *IPTPS 2005*, pp.127–140 (2005).

[28] VMware, Inc.: *VMware Virtual Appliance Marketplace, Virtual Applications & Cloud Computing* (2009).

[29] Liao, X. Jin, H., Liu, Y., Ni, L. and Deng, D.: AnySee: Peer-to-Peer Live Streaming, *IEEE INFOCOM 2006*, pp.1–10 (2006).

[30] Zhang, X., Liu, J., Li, B. and Yum, P.: CoolStreaming/DONet: A Data-Driven Overlay Network for Efficient Live Media Streaming, *IEEE INFOCOM 2005*, pp.2102–2111 (2005).

Tsuyoshi Hisamatsu received a Bachelor of Environment and Information Studies degree from Keio University in 2004. He received Master's Degree in 2006 from Graduate School of Media and Governance and is now a Ph.D. candidate at the same university. His current research interests include overlay network and multimedia streaming. He is a member of IEEE, IEICE, and WIDE Project.

Hitoshi Asaeda is an Associate Professor of Graduate School of Media and Governance, Keio University. He received a Ph.D. in Media and Governance from Keio University. in 2006. From 1991 to 2001, he was with IBM Japan, Ltd. From 2001 to 2004, he was a research engineer specialist at INRIA Sophia Antipolis, France. His research interests are IP multicast routing architecture and its deployment, dynamic networks and streaming applications. He is a member of ACM, IEEE, IEICE, and WIDE Project.