A Streaming Method with Trick Play on Time Division based Multicast

Yoshitaka Nakamura[†], Kazuki Murakoshi[‡]Yoh Shiraishi and Osamu Takahashi

[†]School of Systems Information Science, Future University Hakodate, Japan
[‡]Graduate School of Systems Information Science, Future University Hakodate, Japan
{y-nakamr, g2110045, siraisi, osamu}@fun.ac.jp

Abstract - VoD is a leading service in NGN, which requires a low-load distribution method for network load and the distribution server. In addition, the distribution method must be considered Trick Play in order to respond to user requests. In this paper, we propose a distribution method that combines a number of multicast and unicast to get non-received data, and Fast-Forward algorithm. Then simulation results show an effect of our proposal method on the loads of distribution server and network traffic.

Keywords: VoD, NGN, Multicast, Trick Play, Fast-Forward

1 INTRODUCTION

In recent years, Next Generation Network (NGN), which provides reliability and stability as the circuit switched network and scalability and low cost as the Internet, has become popular.

In NGN, voice call, broadcasting, and data transmission are integrated with the high speed and broadband IP network, moreover advanced services are provided with Quality of Service (QoS) control and access control. In addition, development of the service by competition among providers is expected because some interfaces to access NGN are provided for many providers to use NGN. Above all, Video on Demand (VoD), which provides the large-scale and high quality content like a movie by using QoS control, is a useful service. In the VoD service, providers need to respond immediately to the user ' s request called Trick-Play such as fast-forward and rewind. Therefore, a technique called True VoD is used in a general VoD service to achieve Trick-Play. In true VoD, content is delivered from the delivery server to the requested user by unicast. However, there is a subject of the increase in cost of equipment in unicast due to a load of delivery servers and network increase proportion to increase the number of users.

In previous research, Content Delivery Network (CDN) and Peer-to-Peer network based VoD (P2P-VoD) provide load balancing by efficient placing cache servers to keep a copy of content. In these approaches, load balancing of the delivery server and load reduction of the network are achieved by delivering from an appropriate server in a geographical or network place. However, content is delivered by unicast to achieve Trick-Play in these approaches, larger load reduction is not expected. On the other hand, there are some approaches using multicast such as Near VoD to deliver content from the beginning at regular intervals, Fusion-Stream and Neighbors-Buffering-Based VoD (NBB-VoD) unified multicast and unicast. As the server delivers with multicast in these approaches, a load on the server and network is decreased greater than True VoD. However, There are some subjects such as corresponding to Trick-Play and definition of the number of channels on multicast, connection stability owing to user ' s action as join or leave P2P network, and a load on network owing to a long time unicast.

In this paper, we propose NBB-VoD based streaming method to decrease a load of delivery servers by limited to three connections, which are a multicast to receive from delivery server, a unicast to receive unbuffered data from a user who joins the multicast just before, and a unicast to send a user who joins the multicast immediately after, on Set Top Box (STB) that is a receiving device. Also, we propose an algorithm for Trick-Play, so we achieve more on-demand content delivery method. Furthermore, we compare our proposal method with CDN and Fusion-Stream by Network Simulator, as a result, this paper shows the usefulness of our proposal method in high-quality content delivery service.

2 RELATED WORK

2.1 Content Delivery Network

A Content Delivery Network (CDN) consists of two components, the origin server where the content to be distributed over the Internet is originally stored and cache servers where the content is duplicated and delivered. Multiple cache servers placed distributedly on the Internet, contents are deliverd from the cash server which is the nearest to the user who required delivery. For metrics of the distance to the cash server, geographical distance or distance in network topology or delay time from delivery request to deliver is used. In CDN, the traffic amount is reduced by shortening the distance of using unicast communication. And, network load and processing load on the single delivery server are distributed by locating a cash server distributedly. In addition, user authentication and easy accounting management are also characteristics of CDN, because the service provider manages point-to-point communication. However, with the increase of the users, the increase of the equipment cost and the traffic amount become the problem because of using unicast communication.

2.2 Fusion-Stream

Fusion-Stream is a contents delivery method that combines multicast communication and unicast communication. The contents are delivered from the begining by multicast, and when a user join the session halfway, the non-acquisition part of contents are delivered by unicast from the user who participated just before. In Fusion-Stream, the network load can be reduced greatly in comparison with CDN, because the delivery server performs only multicast delivery and management of IP address and the request order of the user who requests to watch contents. The data transmission to the user who joins next is realized by buffering the data which each user terminal already received. Therefore, the time to watch the contents from the request of contents becomes short, and users can playback contents in any timing. However, as much as an elapsed time of the delivery becomes long, the time of unicast communication between users becomes longer, and the network traffic quantity increases. And there are some problems with non-correspondence to Trick-Play and problems with stability at the time of users' frequently participation and secession of the network.

2.3 Neighbors-Buffering-Based VoD

NBB-VoD is also a contents delivery method that combines multicast communicationto deliver a content with multiple channels and unicast communication to acquire non-acquisition data from other users like Fusion-Stream. In NBB-VoD, the contents data is delivered from the beginning every period of time by multicast. As much as possible users participating in contents delivery deliver non-acquisition data for the user who requests to watch contents by unicast communication. In NBB-VoD, the load of network bandwidth is reduced, because the long-time unicast communication that becomes the problem in Fusion-Stream is divided into the number of channels. And, there is the advantage that the management of the certification and the charging is easy because NBB-VoD uses each STB as a receiving terminal. However, the load of the delivery server and the load of network bandwidth are in a relation of the trade-off because the load of the delivery server increases by the number of channels. In addition, there are some problems in NBB-VoD, including the number of the channels of effective multicast being unknown, the equitableness of electricity which users incured and the network bandwidth, the correspondence to Trick-Play and the point that unicast communication produces from delivery server when contents are delivered by more than the number of the maximum unicast connection of each user.

3 PROPOSED METHOD

In this paper, we propose a delivery method that can reduce the load of delivery servers and network bandwidth in contents delivery than NBB-VoD, by limiting the communication of each STB which is the receiving terminal of users to three. And, we clarify the appropriate number of multicast channels that is not clarified in NBB-VoD by implementing proposed method on a network simulator and evaluating quantity of traffic of the whole network. Furthermore, we propose a control method by the change the communication session of STB from delivery servers and realize Trick-Play.

Fig.1 shows a communication channel between delivery server and STB in the proposed method and NBB-VoD. The delivery server delivers contents using three multicast channels in NBB-VoD(Fig.1(a)). At the same time, multiple unicast channels for the delivery server are necessary in NBB-

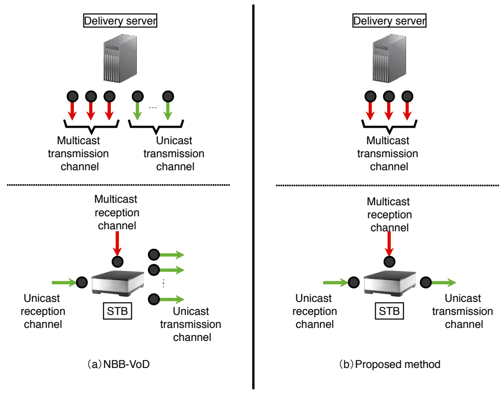

Figure 1: Communication channel of delivery server and STB

VoD, because the delivery server delivers non-acquisition part of data which cannot be delivered by the users who had already joined the session to users who joined the session halfway. In addition, multiple unicast channels are also necessary for STB, because STB performs unicast communication as much as possible. On the other hand, the proposed method(Fig.1(b)) is more efficient because the delivery server delivers contents using only three multicast channels and STB also needs only one unicast channel.

3.1 Contents Delivery

The proposed method is a contents delivery method. This method sets dividing points in contents at regular intervals. And it combines multiple multicast communication to deliver contents from the beginning in each dividing point and unicast communication to transmit and receive non-acquisition part of data for STB. Each STB buffers all the received data until the multicast delivery that its users join is finished. At this time, STB may buffer the one whole content at most. Therefore the storage which can buffer one whole content at least is necessary for each STB. In addition, two network channels for the reception and one network channels for the transmission are necessary for each STB.

Fig.2 shows a delivery example in the proposed method. In this example, we assume the number of partitions (the number of dividing points) 2, and the contents of 30 minutes are delivered by two multicast channels. Therefore the user who request the delivery by 15 minutes later since the first multicast session is started (until half of the content has been delivered) joins the first multicast session. And the user who requested the contents delivery after 15 minutes by 30 minutes later joins the second multicast session. STB 2 is the terminal which request the delivery 5 minutes later since STB 1 had requested and does not have the data from the begining to 5 minites which had been deliverd before joining the multicast session. Therefore STB 2 acuires the data between 5-30 minutes by multicast communication. At the same time, STB 2 acquires non-acquisition data of 0-5 minutes by unicast communication from STB 1 which joined the multicast session

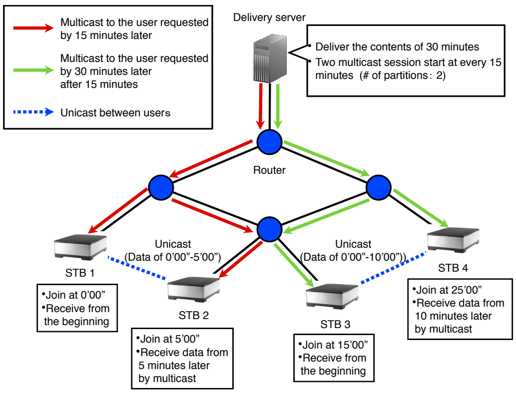

Figure 2: Delivery example in the proposed method

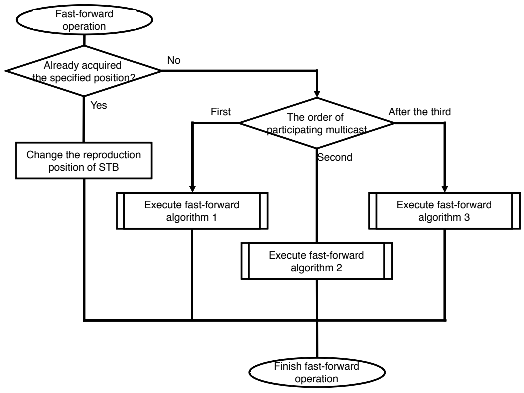

Figure 3: Overview of fast-forward algorithm

just before. Similarly, STB 4 acuires the data between 10-30 minutes by multicast communication, and STB acquires non-acquisition data of 0-10 minutes by unicast communication from STB 3 which joined the session just before. The proposed method uses SIP(Session Initiation Protocol)[4] as delivery requests and reply. And the necessary infromation for the session establishment is described in a form of SDP（Session Description Protocol）[5].

3.2 Type of multicast

The proposed method uses two kinds of multicast of static multicast and the dynamic multicast for normal delivery and Trick-Play. This method uses static multicast to deliver the contents from the beginning to the end by multiple channels in the normal delivery. The static multicast is communication to deliver to the user who request the contents by channels assigned every time interval set beforehand.

On the other hand, in the Trick-Play when some kind of requests of contents such as the fast-foward operation occurred for the data which are not buffered in STB, multicast channels are constructed to deliver the data from the position that user specified to the position buffering in STB. And Trick-Play uses these multicast cannels as dynamic multicast. If the specified position has been already delivered as dynamic multicast, thereare some users who joined the dynamic multicast just before that. In this case, the user who request the contents delivery joins the delivered dynamic multicast session. If there is no participating users and the planned delivery are finished, dynamic multicast session is stopped, because the dynamic multicast is made as a temporary delivery session.

3.3 Fast-forward algorithm

The fast-forward operation often specifies a playback position of non-acquisition part of data in Trick-Play. Fig.3 shows the summary of the fast-forward algorithm in the proposed method. It is necessary to consider two cases in the proposed method, because specified position of fast-forward operation has been already acquired or is not acquired. When the specified position has been already acquired, the proposed method only changes the play back position of output from STB to the playback devices, andfast-forward can be operated without any other special processing. Therefore the contents can be played back with continuing transmission of multicast and unicast. When the data of the specified position of fast-forward operation are not acquired, the users who joined the session just before that and the users who have been joined other multicast buffer the data of the part of contents. Therefore we consider three types of cases. At first, there is a case that a user who want to fast-forward join the multicast session which is started first. Next, there is another case that a user who want to fast-forward join the multicast session which is secondly started (only one multicast session has been started). Finally, there is the other case that a user who want to fast-forward join the multicast session which is tarted after the third. (more than two multicast session has been started before).

The case of participating in the first multicast session When a user specifies non-acquisition part of data and fast-forwards them, we pay our attention to the data buffered by the user who join just after and just before that. If the user who joined just before that buffers non-acquisition part of data, a user who want to fast-forward can receive non-acquisition data by unicast from the user who joined just before that. Therefore fast-forward is realized by switching from unicast session receiving non-acquisition data to unicast session to receive data after the fast-forwarding from the user who joined just before that. But, the reception of non-acquisition data stops by switching sessions. If there is the user who joined just after that, and the unicast transmission to the user is continuing, the data buffered by the user who want to fast-forward has non-acquisition part of data, and unicast session cannot continue(Fig4).

In the proposed method, if the unicast session cannot continue to the user who join just after or if the user who joins just before does not buffer non-acquisition data, dynamic multicast channel is constructed to deliver the data D_{before} from the position where the user specified to fast-forwarding. And the user who want to fast-forward stops transmission and re-

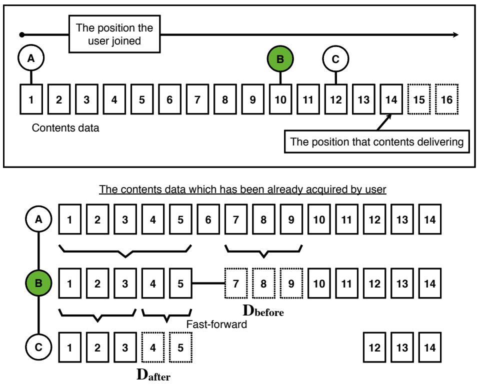

Figure 4: Overview of fast-forward algorithm

ception of unicast channel, and contents data is received from static multicast and dynamic multicast. The user who joined just before that switches a sessions of the unicast, because a user joined just after that transmits non-acquisition part of data D_{after}.If there is dynamic multicast including the position which has been already specified, contents data is received by unicast channel from a user participating in the dynamic multicast. By above mentioned processing, fast-forwarding can be realized without stoping the delivery of itself and anteroposterior user. On the other hand, if unicast session can continue or if a user does not exist just after that, a user who want to fast-forward switches receiving unicast sessions, and fast-forwarding is achieved without stopping delivery. The above-mentioned structure is called "fast-forward algorithm 1".

The case of participating in the second multicast session If a user who want to fast-forward specifies the position of fast-forwarding within multicast session which the user joins, "fast-forward algorithm 1" is used. On the other hand, if the user specifies the fast-forwarding position across the range of multicast which the user joins, the position may be within the multicast that was started before or in the non-acquisition data range by all the other multicast. If the position is in the non-acquisition data range, the delivery server generates dynamic multicast to deliver from the specified position and switches a session. By the multicast session started before, new participation in multicast does not occur, and unicast sessions between users are finished. If there are the users who have already finished unicast session, the data between the positions that participated in multicast session and the position specified by fast-forwarding is acquired using unicast session from the user who participated in multicast the earliest among those users. If all users continue unicast session, dynamic multicast channel is constructed and the data between the position specified by fast-forwarding and the position where the user who finishes unicast session the earliest finishes the unicast session is delivered. The user leaves the multicast session that the user itself participates in and participates in the multicast session that delivery was started just before that. If the user stopped unicast, the data from the end point of the dynamic multicast to the position that participated in multicast is acquired by unicast. The above-mentioned structure is called "fast-forward algorithm 2".

The case of participating in multicast session after the third If there is the position specified by fast-forwarding in multicast session started more than two session before, all users participating in the multicast more than two session before finish unicast session of the normal delivery, and each user buffers all the data delivered by the multicast session that each user participates in. Therefore the user who want to fast-forward leaves participating multicast session and identifies the multicast whose position of delivering is the nearest to the position specified by fast-forwarding among the multicast session. And the user searches the user who have unicast session in the multicast session. If there is the user who does not have unicast session, the data is received by unicast from the user who participated in multicast first among the users who do not have unicast channel. If all users continue unicast session, former multicast session are searched. And the user participates in the multicast session that there is the user who does not have unicast channel and receives these data by unicast communication. If there is no multicast session, dynamic multicast channel is constructed between the user who finish the unicast session the earliest among the multicast session whose position of delivering is the nearest to the position specified by fast-forwarding and the data is delivered using this multicast channel. If the user stopped unicast, the data from the end point of the dynamic multicast sesssion to the position that participated in multicast is acquired by unicast. The above-mentioned structure is called "fast-forward algorithm 3".

4 PERFORMANCE EVALUATION

To show an effect of the proposed method, we implemented each methods of unicast, multicast and Fusion-stream on network simulator ns-2, and evaluated the load of the network and the load of the delivery servers in each method. We measured traffic by adding up packets to flow through the whole network topology every 1 second. And we evaluated the load of the network by this network traffic. We measured amount of transmission data by calculating the sum of packet size that the delivery servers transmitted during simulation time. Each simulation was carried out three times, and we caluculated the mean of them. Table1 shows simulation parameters.

4.1 Evaluation about the number of partitions

To show the effective number of partitions in traffic and amount of transmission, the number of partitions are changed in simulation. The number of partitions used 12 of 1, 2, 3, 4, 5, 6, 8, 10, 12, 15, 20, 25. The evaluation items are average traffic, maximum traffic and the amount of transmission data in each simulation time and each number of nodes. Table2 shows simulation results of each number of partitions.

Table2 shows that the average traffic increasing though amount of transmission that is the load of the delivery server increases

Table 1: Simulation parameter

Simulation time	$600s$
# of the participation nodes	480
# of the relay nodes	168
Packet size	$1000byte$
Transmission rate	CBR, $8Mbps$
Transmission band of the participation nodes	$100Mbps$
Transmission band of the relay nodes	$1Gbps$
Delay between nodes	$5ms - 100ms$

Table 2: Change of the traffic by the number of partitions

# of partitions	Ave.[MB/sec]	Max.[MB/sec]	Amount
1	1206.6	2373.5	572.1
2	1039.1	1605.4	857.1
3	894.8	1410.3	1140.7
4	833.7	1292.5	1425.1
5	788.6	1321.2	1710.4
6	771.5	1302.2	1994.0
8	757.9	1267.6	2565.2
10	745.1	1372.6	3135.3
12	738.9	1393.2	3703.6
15	747.3	1384.2	4558.2
20	750.0	1464.1	5982.3
25	789.6	1506.3	7406.8

when the number of partitions is more than 15. If the number of partitions is set to 12, the amount of network traffic is minimized, and network is effective in total. However, the performance of the delivery server may be short with the number of partitions 12, because 6.5 times as much as the amount of transmission in comparison with the number of partition 1. In that case, the number of partitions lowered like 10 or 8 as needed.

4.2 Evaluation about traffic and amount of transmission

For the evaluation about traffic and the amount of transmission, the number of partitions of the proposed method is decided with 12 that was effective number of partitions. The evaluation items are average traffic, maximum traffic and the amount of transmission data in each simulation time and each number of nodes. Table3 shows the simulation result of each method. And Fig.5 shows the change of the traffic in the elapsed time of the simulation.

From the result of table2, the proposed method reduces 40.9% of average traffic in comparison with unicast and reduces 38.8% of average traffic in comparison with Fusion-Stream. And from Fig.5, the proposed method, Fusion-Stream and unicast are seen to change in the same way until 50 seconds. However, the degree of leaning becomes small after 50 seconds, because the number of partitions is set to 12 by

Table 3: Result of the traffic and the amount of transmission data

	Ave.[MB/sec]	Max.[MB/sec]	Amount
Proposed method	738.9	1393.2	3703.6
Fusion-Stream	1206.6	2373.5	572.1
Unicast	1250.5	2454.6	136550.4

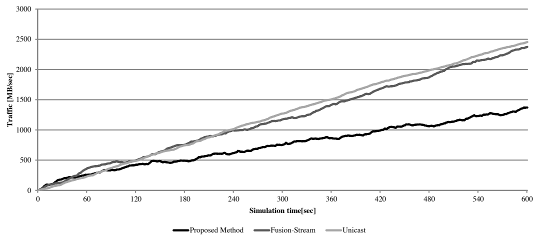

Figure 5: Change of the traffic

proposed method and the second multicast is started at 50 seconds, and the unicast that occurred by 50 seconds is finished. On the other hand, traffic increases in proportion to increase of the number of the participation nodes by the unicast. The change of Fusion-stream becomes similar to unicast, because time for unicast becomes long so that participation in delivery is the latter half. The proposed method reduces 43.2% of maximum traffic in comparison with unicast and reduces 41.3% of maximum traffic in comparison with Fusion-Stream. This result shouw that the proposed method could reduce greatly the peak load of network. In addition, the load of the delivery server is greatly reduced, because the proposed method reduces 97.7% of transmission in comparison with unicast. However, as for the amount of transmission increase 6.5 times in comparison with Fusion-Stream. This is unavoidable in a property of proposed method delivering using multiple multicast.

5 CONCLUSION

We proposed a distribution method to reduce network load by limiting the communication of STB by three based on NBB-VoD system in VoD service over NGN. We made Trick Play possible in this method using the algorithm in consideration of the situation assumed especially about fast-forward. We implemented the proposed method on network simulator, and evaluated performance with the existing methods about the number of multicasting division, the amount of traffic and the amount of transmission. The evaluation shows that our proposed method can reduce the load to network bandwidth or distribution server greatly compared with the existing method.

For future work, we plan to implement Trick-Play algorithm on network simulator, and evaluate in more realistic environment. We also make improvements of the algorithm considering the delay time from making the request of the trick play until switching sessions and receiving the content data.

REFERENCES

[1] B.Krishnamurthy, C.Wills, and Y.Zhang, On the Use and Performance of Content Distribution Networks, Proc. of ACM SIGCOMM Internet Measurement Workshop 2001, pp. 169–182 (2001).

[2] V.Janardhan, and H.Schulzrinne, Peer assisted VoD for set-top box based IP network, Proc. of ACM Peer-to-Peer Streaming and IP-TV Workshop 2007, pp. 335–339 (2007).

[3] T.Taleb, N.Kato, and Y.Nemoto, Neighbors-Buffering-Based Video-on-Demand Architecture, Signal Processing: Image Communication, Vol.18, Issue 7, pp. 515–526 (2003).

[4] J.Rosenberg, H.Schulzrinne, G.Camarillo, A.Johnston, J.Peterson, R.Sparks, M.Handley and E.Schooler, SIP: Session Initiation Protocol, RFC3261 (2002).

[5] M.Handley, V.Jacobson, and C.Perkins, SDP: Session Description Protocol, RFC4566 (2006).