

Buffermap adaptation method for MP2P multimedia streaming protocol

Wook Hyun*, ChangKyu Lee*, ShinGak Kang*, Juyoung Park*

*ETRI (Electronics and Telecommunications Research Institute), Korea

whyun@etri.re.kr, echkyu@etri.re.kr, sgkang@etri.re.kr, jypark@etri.re.kr

Abstract— On providing multimedia stream over a mesh-based overlay network, the starting point of buffermap is determined through negotiation with other peers. When it boots up, it does not have a starting point of its buffermap, and it can determine this value only by negotiations with other arbitrary peer. If it selects a bad start point, the peer cannot receive stream data because other peer's buffermap may not have a fragment that it needs in some time. There are some jitters in processing media for presentation and network delivery, and it accumulates over timer. This leads to the peer lag behind of actual stream. Hence, it is crucial to adjust the start point of buffermap appropriately during streaming. This paper describes how the problematic situation can be happen, and how to adapt the buffermap for stable multimedia streaming service over a mesh-based overlay network.

Keywords— buffermap, adaptation, multimedia streaming, mesh, P2P network, overlay network

I. INTRODUCTION

Nowadays, many global service providers, such as Netflix, YouTube and Amazon, makes use of CDN with their own network infrastructures for providing multimedia streaming services. However, this is not affordable to any service provider that does not have those resources. Peer-to-peer multimedia streaming provides largely scalable services to the massive number of users without bottleneck or load concentration to the comparatively small number of sources. Especially, the quality of video increase the size of video stream and require more network bandwidth. As the quality of video goes to 4K UHD, it needs more network bandwidth, and it is not appropriate to accommodate many concurrent users with the small number of servers.

For providing ITU-T SG11 Q8 is under development of multimedia streaming protocol in managed P2P network that is defined in ITU-T X.609 series [1][2][3][4][5]. Figure 1 shows overall procedures for exchanging sliced fragments with each other. In negotiation phase, they exchange their buffermap that represents the possession of a specific fragment. If the negotiation successful, it goes to streaming phases that exchanges actual fragment that contains media data. In this phase, there are two kinds of relationship can be made; neighbourhood and partnership. In partnership, each

peer sends notification on acquiring new fragment from any other peer. On receiving this notification, the corresponding peer may request to send the fragment. This can reduce unnecessarily frequent buffermap exchanges.

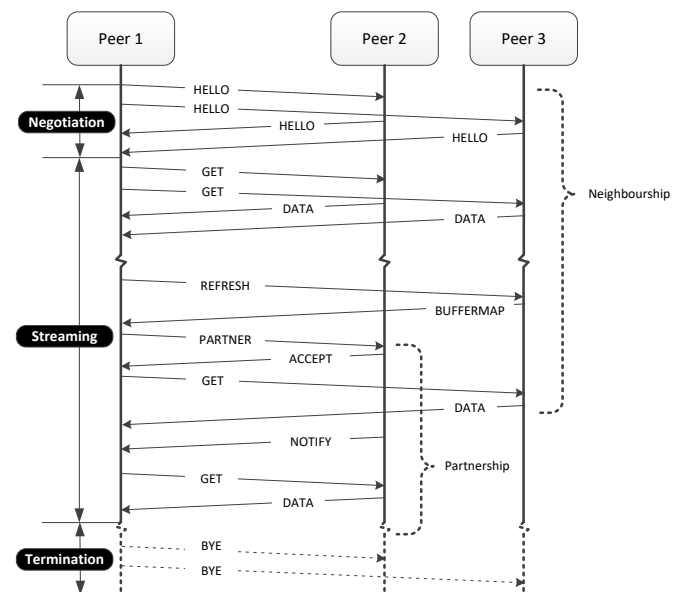


Figure 1. Peer protocol for multimedia distributed streaming defined in ITU-T X.mp2p-mspp

In P2P multimedia streaming, each peer has their own buffermap that represents the status of possession of fragmented stream data. Peers exchange their buffermap with other peer during negotiation phase. Figure 2 shows the internal structure of buffermap defined in ITU-T X.mp2p-mspp[6]. The SP points to the identifier of the starting fragment of peer's buffermap.

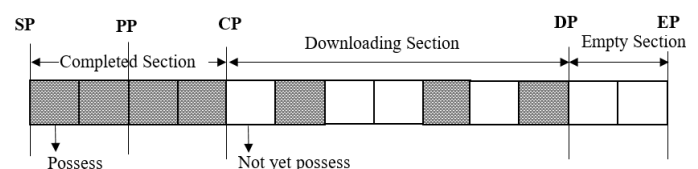


Figure 2. Structure of buffermap defined in ITU-T X.mp2p-mspp

Other points, such as PP, CP, DP and EP, are used for internal flow control or exchanges of fragmented stream data. Each peer has different shape of buffermap with other peer, and newly joining peer must get start point of its buffermap from other peer since OMS(Overlay management server) does not provides it.

This paper describes the problematic situation that can be happen on mesh-based multimedia streaming in clause 2, and explains the proposed buffermap adaptation method in clause 3. In clause 4, we will explain implementation of proposed method, and conclude this paper in clause 5.

II. NECESSITY OF BUFFERMAP ADAPTATION

In P2P multimedia streaming services, only one media source generates sliced fragments from media source and push them into overlay network. In the overlay network, each peer exchanges the data with each other. Unlike tree-based streaming, each peer holds their own buffermap and uses it for negotiation with other peer, and pull a specific fragment in parallel. Each peer constructs their buffermap, and all peer has different shapes. Especially, the starting point of buffermap is most important point because it decides the position of stream.

During negotiations, if it sets the starting point close to the peer's CP, it can comparatively real-time stream, but it may lead to frequent buffering. If it sets the starting point close to the peer's SP, it can minimize start up delay, but it may not get required fragment on time since its buffermap will fall behind from the streaming.

Figure 3 depicts the situation P2 cannot get fragment from P1 even though P1 has fragments that P2 will need in near future. In Figure 2, the grey box means that the peer has the fragment, and white box means that the peer does not have the fragment. Unlike the file sharing services, each peer has limited interval of buffermap of the entire streaming since it is hard to know the end of the streaming. Hence, the buffermap acts like a sliding window, and it goes to right over time. If P2 has a buffermap that fall behind requests a specific fragment, P1 cannot provide it, and this situation will happen with other peers as well.

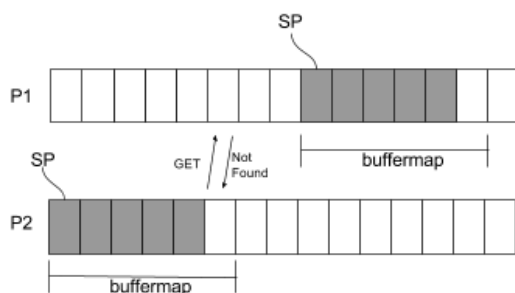


Figure 3. Exchanges failure due to the buffermap lagging behind

If the bitrate of the video is high and the size of fragment is relatively small, it accelerates the generation rate of fragments. This leads to accelerated movement of sliding buffermap. This makes this situation more often. If the peer's buffer does not move to the actual video rate, the peer becomes orphaned in the overlay network and no longer obtains new data from another peer.

In addition the orphaned situation, it will arise the risk of infection of this situation to the overlay network. As is, if a newly joining peer decides the location of its buffermap on interacting orphaned peer, it will have same situation and other peers also has infected location of buffermap as shown in Figure 4.

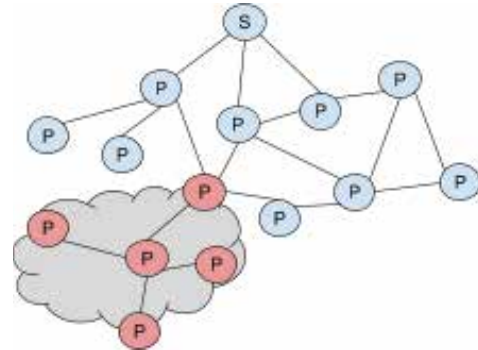


Figure 4. Infection of orphan peer of newly joining peer

The peers within a grey cloud are orphan from the overlay network if there is no appropriate adaptation. Even though the orphan interacts with active peers, it will release the interaction since the active peer does not have fragments that the orphan peer needs, but the buffermap of orphan peer does not move.

Therefore, it is crucial to set optimal starting point of buffermap, and it needs to adapt it if it figures out that it begins to lag behind. It is also possible to use this feature to synchronize the playback time of each peer device.

III. PROPOSED BUFFERMAP ADAPTATION METHOD

The proposed method is for dynamically adapt the buffermap based on the performance of the peer. It means that this method readjusts the location of buffermap based on the situation. If the network is quite good, it can put the buffermap close to the live streaming, and it will put it for delay streaming, if not. That is, it readapts the buffermap based on average download rate and video play rate.

If the buffer map of the current negotiating peer is ahead of its buffer map and there is no intersection, move the starting point (SP) of the peer forward.

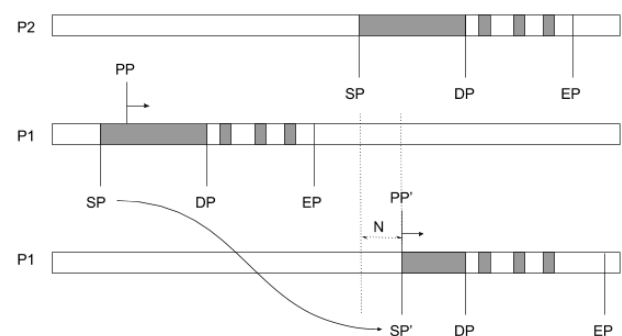


Figure 5. Buffermap adaptation for smooth delayed play

As shown in Figure 5, the buffermap of P1 is fall behind of the buffermap of P2. That is, the SP of P2 is larger than the EP of P1, and there is no intersection between two buffermap. In this case, P1 begins to readapt its buffermap by moving its start point to SP' closed to the SP of P2 for the smooth delayed play rather than more live play.

In order to set the location of buffermap for live streaming, the peer sets its new starting point to close to CP as shown Figure 6.

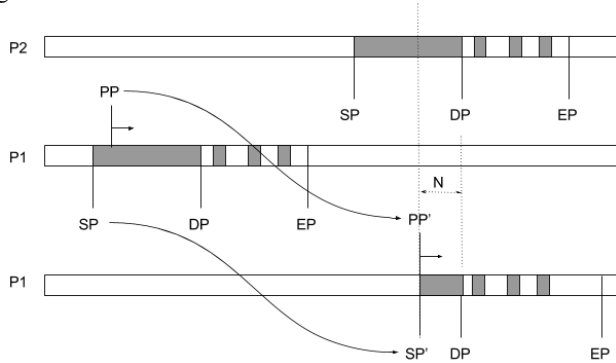


Figure 6. Buffermap adaptation for live play

However, if it sets too much close to CP or SP of corresponding peer, it may fail to keep receiving fragments for newly adjusted buffermap, because when a requesting peer request M fragment to corresponding peer, the SP of corresponding peer may already incremented to M+1. In this case, the peer still cannot receive new streams even though it readjusted its buffermap. Accordingly, if the adjusted SP values is too close to the CP of the corresponding peer, it may suffer frequent buffering in play of the stream. This is reason why there may be small number of peer has the fragments. Hence, it needs to give some margins.

On adjusting the start point of the buffermap, the peer makes use of the average download rate (ADrate) and the average play point move rate (APMrate). The PP moves whenever the media player fetches a fragment for presentation. Actually, APMrate can be approximately same with the generate rate of fragments. Unlike file sharing, the number of origin media source will be just only one, and the origin generates fragments for distribution. The average download rate is important factor for deciding the location of buffermap. If it is low, the buffermap will be located close to peer's SP with some margin, since the possibility that a peer has a specific fragment will be higher if it is close to peer's SP. The combination of these two values can determine the optimal value best suited to the multimedia characteristics of the session in consideration of the network situation, peer device performance, and the like.

On determining the start point of the buffermap, there are two cases; initial and readjustment. The peer determines initial point without considering download rate or play rate since it does not have that information. If the peer is just got initiated and no SP value is specified, the initial SP value is determined as follows;

$$mySP = peerDP -$$

$$ROUND[\frac{1}{2} * max\{ (peerDP - peerSP), 1 \}]$$

In the case of readjusting the start point of buffermap, it checks whether myEP is smaller than peerSP. If so, it readjust the value as follows;

$$R = APMrate / ADrate; /* (0 < R < 1) */$$

$$N = ROUND[max\{ (peerDP - peerSP) \times R, 1 \}];$$

$$\text{if } (ADrate > Limit) \quad mySP = peerDP - N;$$

$$\text{else} \quad mySP = peerSP + N;$$

When the average download speed is higher than the limit value, it is decided at a point close to the peerDP. If the average download speed is lower than the limit, the value is determined at a point close to the peerSP, so that the stability and real-time characteristics of the service can be flexibly selected. It is possible to calculate the limit value by use of APMrate.

IV. IMPLEMENTATION/EXPERIMENTS OF BUFFERMAP ADAPTATION AND FUTURE WORKS

We have added this feature to our implementations [7][8], and it solves orphan and infection problems. In the experimental, we used 4K UHD video clips with maximum 60Mbps bitrate, 30FPS over local gigabit network, and it works smoothly. However, there are some flash jump or degrade of video quality on buffermap readaptation, since the source does not consider video frames on creating the fragments.



Figure 7. Implementation and experiments of p2p-based multimedia streaming with buffermap adaptation in small scale

Figure 7 shows a screenshot of implementation, and the left two windows are for live streaming and the right pane shows an actual status of download/upload rate in real-time.

As a future work, we will put more peers in the heterogeneous network by using emulation software such as CORE, and enhance this method for smooth adaptation without flash jump.

V. CONCLUSIONS

This paper describes orphan and infection problems that can happen on multimedia streaming services over mesh-based overlay network. The sliding buffermap is tightly dependent on the starting point that decides the location of

buffermap. The buffermap moves over time on receiving new fragment from other peers, but it cannot receive new ones if it is located in wrongful location. Furthermore, it will be infectious to other newly joining peers. This paper proposes a method for adapting buffermap by using the average download rate and the average play point moving rate. We have implemented this features to our implementation, and confirmed that it works well. However, we also find that it needs more improvement and large-scale emulation for better smooth and real-time streaming in the future.

ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No. 2015-0-00242, Development of standards for distributed UHD streaming protocol).

REFERENCES

- [1] *Managed P2P: Framework*, ISO/IEC TR 20002, 2013
- [2] *Managed P2P Communications: Functional architecture*, ITU-T Std. X.609, 2014.
- [3] *Managed P2P Communications: Peer activity management protocol*, ITU-T Std. X.609.1, 2016.
- [4] *Managed P2P Communications: Overlay resource control protocol*, ITU-T Std. X.609.2, 2016.
- [5] *Managed P2P Communications: Multimedia streaming signaling requirements*, ITU-T Draft Std. X.609.3, 2016.
- [6] *Managed P2P Communications: Multimedia streaming signaling protocol*, ITU-T Draft Std. X.mp2p-mspp, 2016.
- [7] W. Hyun, SH.Kim, CK.Lee, SG.Kang, "Implementation of PREP (P2P-based streaming Protocol) Client" in Proc. ICACT 2012, pp 628-631, 2012.
- [8] W.Hyun et al, "Study on Architectural Model and Design of P2P-based Streaming Client", ICHIT 2011, CCIS 206, pp.1-7, 2001.



Wook Hyun is a research staff member with ETRI (Electronics and Telecommunications Research Institutes) since 2000. He has received M.S. degree in Information Communication Engineering from Chungnam National University, Korea in 2000. His research interests include VoIP, SIP, NGN, P2P, overlay networking and digital signage.



Ju Young Park is working for ETRI from when he has received his PH.D degree in 2001 from Chungnam National University. Thereafter, he took project editorships both in ITU-T and ISO/IEC/JTC1, and he also has developed three International Standards (IS). His major research areas are smart work, Multicast, QoS protocol and architecture. He also has great concerns on mobile communication and IOT.



Changkyu Lee received B.S. degree in Computer Science and Engineering from Inha University, Rep. of Korea, in 2008. Since 2008, he has been an integrated M.S. and Ph.D. student of University of Science and Technology at Electronics and Telecommunication Research Institute Campus, Rep. of Korea. Since 2017, he has been working for Electronics and Telecommunications Research Institute. He is actively participating in international standards bodies such as ITU-T SG11 and SG16, and ISO/IEC JTC1/SC 29/WG11 (MPEG), IEC TC100, etc. His main research interests include multimedia communications and applications, ICT converged services, and content distribution.



Shin-Gak Kang received B.S. (1984), M.S. (1987) in Electronics Engineering, and Ph.D. (1998) in Information Communication Engineering from Chungnam National University, Rep. of Korea. Since 1984, he has been working for Electronics and Telecommunications Research Institute, Daejeon, Rep. of Korea. Since 2008, he has been a professor in the Department of Information and Communication Network Technology of University of Science and Technology. He is actively participating in various international standard bodies as a Vice-chairman of ITU-T SG11, Convenor of ISO/IEC JTC1/SC6/WG7, etc. His research interests include multimedia communications and Applications, ICT converged services, contents networking, and Future Network.