

# A New Approach for the Construction of ALM Trees using Layered Video Coding

Yohei Okada, Masato Oguro, Jiro Katto, Sakae Okubo

Graduate School of Science and Engineering, Waseda University

3-4-1 Okubo

Shinjuku, Tokyo 169-8555, Japan

{okada,masato,katto}@katto.comm.waseda.ac.jp, sokubo@waseda.jp

## ABSTRACT

This paper presents a novel approach for efficient tree construction in ALM (Application Layer Multicast) using layered video coding. This mechanism is suitable for the users in heterogeneous environment such as xDSL and CATV, where upload and download speeds are different. We point out some problems in these environments of the ALM and show their solutions at two stages. First, we redefine the “degree parameter” which originally specifies the number of single rate streams the host can transmit, to reflect asymmetric link properties and to accommodate layered streams. Second, we develop an efficient tree construction algorithm using the renewed degree parameter, which maximizes total throughput of all hosts. Simulations using NS-2 prove improvement in throughput, delay and overhead by our proposal. Software implementation is also carried out, and subjective quality improvement is verified.

## Categories and Subject Descriptors

C.2.1 [Network Architecture and Design]: Distributed networks

## General Terms

Algorithms, Performance, Experimentation

## Keywords

Application Layer Multicast, Overlay Network, Layered Video Coding

## 1. INTRODUCTION

As network evolves into “broad-band”, multimedia contents like music and video tend to spread over the internet. In the conventional client/server model assuming multiple unicast connections, however, some problems are inevitable such as network congestion around the server or on bottleneck links and

too much burden on the server due to large content sizes and a huge number of clients. IP multicast has been thought to be the most valid system that solves these problems completely [1]-[3], but its deployment is quite slow due to the complexity of route control, necessity of replacing routers for multicasting and so on.

Instead, ALM (Application Layer Multicast) emerges as an alternative to the IP multicasting, in which each host executes route control and packet forwarding not on IP layer but on application layer. A video stream transmitted from the source is received, played by some hosts and forwarded to other hosts simultaneously. There are some merits such as load-balancing and cost-down because ALM does not need any high performance servers.

On the other hand, however, ALM also has some issues to be solved. For example, hosts which are located on a lower-position of overlay network (or ALM tree) can not receive data when its higher-position hosts leave the tree suddenly. In addition, when layered coding is applied, receivable layers by the host might be reduced due to limitation of its higher-position hosts. Therefore, it is important how to construct the ALM trees and to adapt to dynamic behavior of the ALM trees.

To construct ALM trees, delay and bandwidth are popular metrics utilized in general [4]-[9]. For interactive communication like IP video phone and TV-conference, much delay causes fatal performance degradation and the delay metric should be applied. For one-way-broadcasting like video streaming, the bandwidth metric should be utilized to achieve to receive high delivered quality of contents [4]-[6].

We propose an approach to construct the ALM trees for layered video streaming. Thus, we give priority to bandwidth over delay. In this case, we consider that the validity can be demonstrated only by making each host receive at various rates according to their environment and by transmitting not a single rate but multiple rates such as in layered coding. Our goal is to make total throughput of all hosts which are in various environment to be maximized by using layered coding.

Moreover, we consider the problem of dynamic behavior of the ALM trees such as parent hosts’ departures. When a host in higher-position leaves the tree, its descendants cannot receive the data. Thus, they need to find new parents that are not affected by the departing of the host. Instead of simply re-joining the tree, we propose an efficient process that shortens the recovery time and saves the number of control packets.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

*P2PMMS’05*, November 11, 2005, Singapore.

Copyright 2005 ACM 1-59593-248-8/05/0011...\$5.00.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 points out the issues of ALM trees and our solution idea. Section 4 describes the proposal method in detail, and Section 5 shows its validity by simulation and implementation. Finally, Section 6 concludes this paper.

## 2. ISSUES AND OUR SOLUTIONS

There are mainly two issues of the ALM using layered coding. First, a child host can not receive the rate that is more than the rate its parent host receives at due to characteristics of streaming technology [5]. Thus, in case of Figure 1, host B which is a child of host A is not able to receive at high rate even if B has rich bandwidth to download data because A receives at poor rate. In such a case, layered coding is not used adequately and each host must choose its parent host carefully. The second issue is complexity to find a proper parent host [6]. Usually, a new participant host joins the ALM tree as a leaf host. Then it searches for its neighbor hosts by trading their bandwidth information and find an adequate host that forwards stream data at the rate which it desires. However, this process is complex and takes much time. Still more, it may be difficult to attain this end host in an optimum manner because it can get only local area information of the overlay network once it joins the tree.

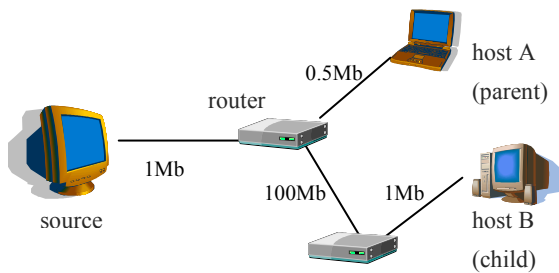


Figure 1: Improper relation between hosts A and B.

Our solution of these matters is to use “degree” parameters. This parameter originally specifies how many hosts a parent host can maintain [10]. The value depends on the parent’s forwarding capacity. For example, when the forwarding capacity of host  $i$  is  $F_i$  and the single streaming rate is  $R$ , host  $i$ ’s degree  $D_i$  is calculated as follows;

$$D_i = F_i / R \quad (1)$$

We will redefine this parameter in the next section for multiple streaming rates so that each host can decide its parent properly. One of the advantages of using this is that the value of parameter is known in advance. Thus, each new participant can decide its parent host when it joins the ALM tree and there is not necessity of looking for an adequate parent later.

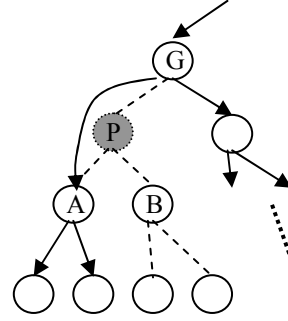


Figure 2: Recovery phase caused by the leaving of host P

Then, it is obvious by the degree parameters that not all descendant hosts decide their new parent hosts easily when the parent host leaves. In Figure 2, all hosts’ degrees are 2, host B (or host A) cannot reconnect to host G which is a parent host of departing host P, and it must search for another host which has an unused degrees. The simplest way to do that is to rejoin as a new participant host. However, it is considered that this takes a lot of recovery time and overhead. Therefore, we have devised a new method for efficient recovery of the tree.

## 3. PROPOSAL MECHANISMS

### 3.1 Redefining the degree parameters

On the Internet, bandwidth of access link, especially up-link, between an edge router and each host of that area is generally much narrower than that of the trunk link. Thus, it is considered that up-link of a sender host is to be the bottleneck link on peer-to-peer networking [6][14]. Based on this, the degree constraint parameter which represents how many hosts the parent host can have is defined as (1). It is easy to decide whether a new participant host can join the parent host or not by referring to this parameter. In ALM using layered coding, the streaming rate  $R$  is not a fixed value and we should redefine it.

At first, we use  $R_1$  which represents the rate of the base layer stream instead of  $R$  in (1).

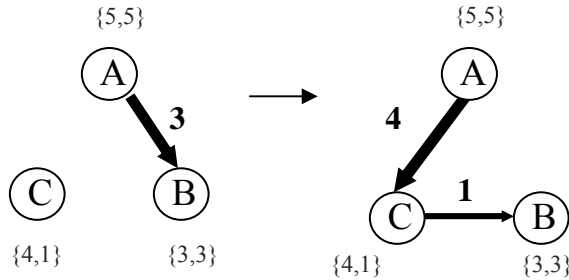
$$D_i = F_i / R_1 \quad (2)$$

Second,  $R_1$  is assumed to be ‘1’, and the ‘accumulative’ rate of each layer’s stream is given by real number proportional to  $R_1$ . As a result, the new degree represents the number of streams which the parent host can have. For example, when rates of multiple streams are {50, 50, 100} (kbps) and host  $i$ ’s sending capacity is 300 (kbps), the degree of the host is 6 according to (2), and the ratio of accumulate rates is 1:2:4. In such case, this host  $i$  has streams of ‘4 base layers and 1 second layer (1\*4+2\*1=6)’ or ‘1 second layer and 1 third layer (2\*1+4\*1=6)’ and so on. We call this degree parameter ‘out-degree’. Moreover, this approach applies to the receiving capacity of the hosts. That is, the number of layers which the child host desires is shown by a real number. We call this degree parameter ‘in-degree’ to distinguish from the out-degree.

Finally, we add an important constraint (incentive) to the relation of ‘in’ and ‘out’ degrees. In ALM video streaming system, each host contributes a portion of the bandwidth to the overlay in exchange for the receiving service. A host which is able to forward little data should not receive much. Otherwise, other hosts have less streams in return for satisfaction of the host. Thus, any host’s in-degree must be limited equal to or less than the out-degree as follows:

$$\text{out-degree} \geq \text{in-degree} \quad (\text{at each host}) \quad (3)$$

For example, new participant host C is eager to receive the 4th layer but can forward only the base layer in Figure 3. When host C becomes the child of host A instead of host B in order to satisfy his demand, host B is to be the child of host C and can receive only the base layer’s stream. At the same time, descendants of host B can only receive the base layer only, and total throughput of the descendants will be less as a result. Thus, host C’s in-degree should be restricted to ‘1’ in this case.



**Figure 3:** Necessity of the in-degree constraint;  $\{x,y\}$  represents an in-degree and an out-degree respectively, and the bold type does the number of layers

However, oppositely speaking, the bigger a host has an out-degrees (the host can send data at higher rate), the bigger the host has an in-degrees (the host gains throughput from the ALM session). In our proposal method, as described later, such a host can exist in higher-position of the tree, and it has many advantages. For example, the delay from the source become shorter and the affection of departure of other hosts is reduced.

Such an incentive idea is adopted by BitTorrent [15], too. That is a P2P file sharing system in which a user supplying much upload link can download the data at high speed. BitTorrent is popularized worldwide, and it is considered this incentive idea is very effective.

CoolStreaming [16] is an ALM to which BitTorrent system is applied for video streaming. This is not a tree type but a mesh type network, and a host which has a big out-degree has also a big in-degree. Thus, such a host can gain higher quality video from the ALM. However, because each host does not have a fixed parent host, the latency until the video is played back becomes larger.

SplitStream [18] whose scheme uses multiple description coding (MDC) of streaming media also defines resemble parameter in and out-degree. The transmitting rate of one stripe is fixed and each host’s degree is decided simply as how many stripes it can forward and receive. However, in case each stripe’s rate is different such as the instance described example, this parameter should be defined more flexibly as our method. Especially, the high rank enhanced layer streams are generally bigger data size than the low rank layer streams in layered coding [3].

### 3.2 Construction phase

We build the ALM trees by using redefined degree parameters in the tree construction phase. Our goal is to construct a tree which satisfies all hosts’ in-degrees. There are following 4 key points:

- **A participant finds a proper parent when joining the tree**

In order to avoid the problem of complexity that a participant finds its proper parent after it has joined the tree, it should be able to decide the host at the same time when joining to the tree.

- **Parent’s out-degree  $\geq$  child’s in-degree**

To satisfy each host’s in-degrees, its parent must have an out-degree which is bigger than the in-degree of the child.

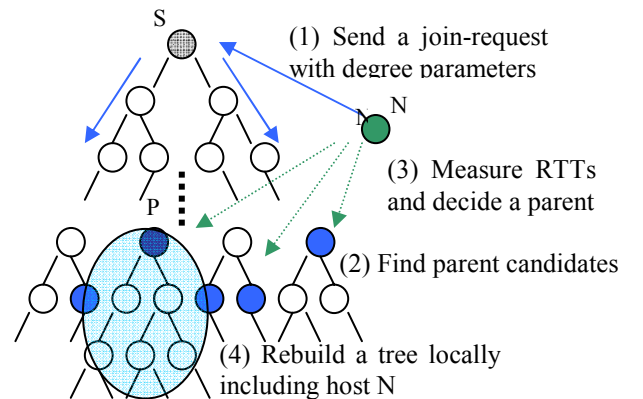
- **Finding not only a parent but its own child**

In relation to the above two points, a participant joins the tree not only as a leaf host but also as an intermediate host of the tree. Thus, it must find its own child simultaneously.

- **Adopting ‘delay’ as the second metric**

To avoid the problem of bandwidth waste by connecting hosts that are far from each other in real network, we use a RTT (Round-Trip-Time) metric, too.

Based on the above key points, we explain the process of constructing ALM trees concretely as Figure 4 shows.



**Figure 4:** Tree construction process

**Step.1: Send a join-request message to the source**

New participant host N sends a join-request message to the source with its in and out degrees information. The source address and degree information are already known.

**Step.2: Search for candidates of the parent host**

All hosts maintain the degree information of itself and its children and grand-children. The host receiving the join-request (at first, the source) refers to its degree information and N’s one, it becomes the parent candidate of host N when each of the following two conditions is satisfied at least. Otherwise, it forwards the request message to its children and this process might be repeated.

- N’s in-degree  $\leq$  its (remaining) out-degree
- N’s out-degree  $\geq$  maximum of its children hosts’ out-degrees

The second condition is based on the concept that hosts which have big out-degrees should be in higher-position of the tree. This is the incentive idea as described previously in Section 4.1. When N’s out-degree is bigger than those of the children hosts, N is in higher-position of the tree and can take more advantage of various points than other hosts.

**Step.3: Decide the parent**

The parent candidates send their response messages to host N and host N measures RTTs respectively. Then, host P which has the minimum RTT is decided to be the parent host of N.

**Step.4: Exchange link connections locally**

Under the condition ① at Step 2, host N is connected to host P simply and N becomes a leaf host. Under the condition ②, P needs to exchange connections in the local area which includes hosts from P to P’s grand-children based on the degree and N can join any host of them. At this time, their total out-degree of them are equal to or bigger than total in-degrees because of (3). Thus, this process can be completed locally.

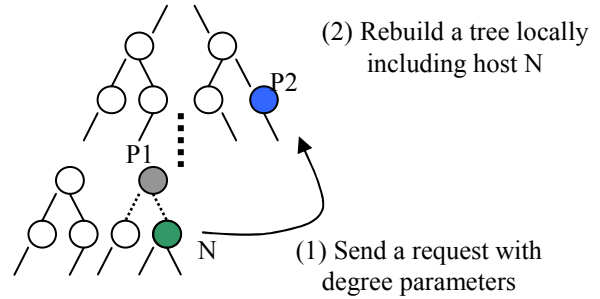
Dagster [19] also builds the ALM tree using up and down link parameters and a new participant can cut into an existing relationship of two hosts and a child is able to have multiple parents. However, transcoding by end host is considered also a heavy task even for new machine and the model hosts’ out-degree is much bigger than the real. Then, we considered multiple parents system is difficult.

**3.3 Recovery phase**

When a node departure happens, the tree recovery process is invoked. Node departures are of two kinds: graceful departure and sudden failure. In the former case, the departing host notices its will to the source and neighbors. The descendants of the host can decide new parents that have unused out-degrees beforehand and reconnect to them seamlessly. In the latter case, on the other hand the departing host leaves suddenly and cannot give his descendants the time to prepare for the reconnection. This may happen due to a computer crashing or congestion of the network links.

In either case, the descendant hosts of the departing host need to find their new hosts. The simplest way is to rejoin the source after being aware of their parent’s departure and decide new

parents as a result of Steps 1~4 as described above. However, we consider that this takes too much recovery time and overhead.



**Figure 5:** Recovery phase, where host P1 leaves trees and its child host N sends a request to host P2 to be a new parent

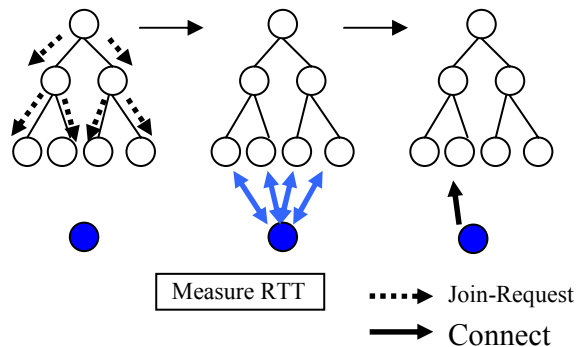
Then, our proposal is that all new participant hosts memorize (cache) their parent candidates which are found at Step 2, and request from one which has the second minimum RTT of them to reconnect directly when their ancestors depart the tree as Figure 5 shows. This new parent host (host P2) also can send data to host N at the rate which host N desires and host P2 is near host N practically, too. In this way, descendants of the departing host only execute Step4 and can shorten the recovery time and overhead.

This recovery scheme does not require that the potential parent hosts have unused out-degrees like the existing method [10] [12] because a participant host can cut into the relationship between the decided parent and its child described at Step 4. Therefore, once the parent candidates are found, they are not alternated unless the potential parent itself leaves the ALM tree.

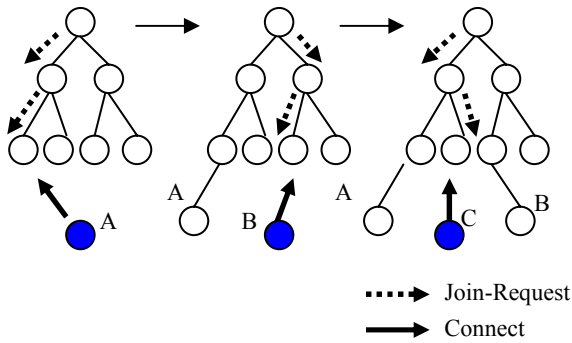
**4. EVALUATIONS**

**4.1 Simulation results**

**4.1.1 Construction phase**

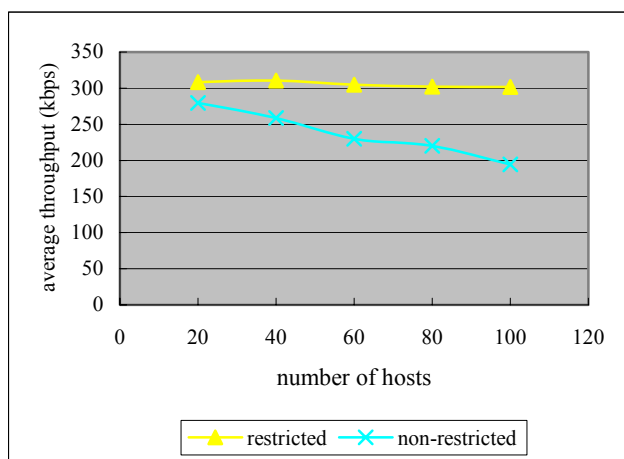


**Figure 6:** RTT method, in which a host with the minimum RTT is chosen as a parent among many candidates.



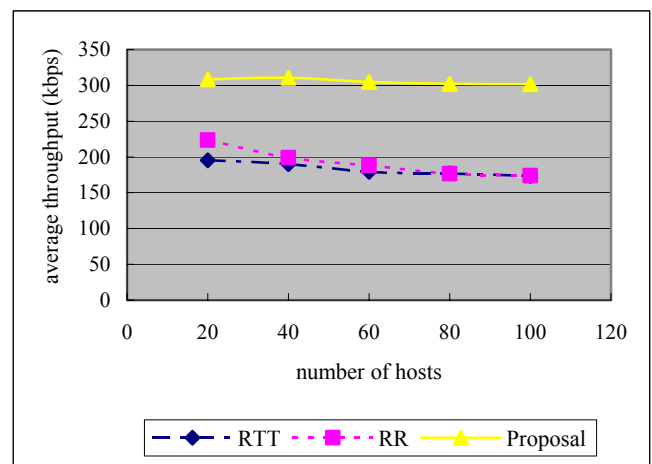
**Figure 7:** Round-Robin method, in which a parent host is chosen in a round-robin manner

We evaluate our proposal method in ns-2 simulator [11]. The streaming rates of each layer are {125, 125, 250} (kbps) by UDP, and the number of hosts of which degrees are assigned randomly is changed from 20 to 100. The link delay between each host is from 5 to 495 (msec), and bandwidth of up-link is from 128 to 1000 (kbps), down-link is equal to or bigger than it. The bandwidth of trunk-link is not considered. Under this condition, we measure three values; average throughput of all hosts, average delay of each link and the number of control packets (overhead). We compare our proposal with two previous methods, RTT method (Figure 6) and Round-Robin (RR) method (Figure 7). In the former method, a join-request is forwarded repeatedly until leaf hosts which have at least one remaining out-degrees will be found, and the participant decides its parent by measuring RTTs of the leaf hosts. In the latter method, a join-request is forwarded to a parent candidate selected in a round-robin manner until a leaf host is found which has at least one out-degree will be found. This method is expected to contribute to load balancing and reduction of control packets. First, we evaluate the validity of in-degree restriction of Eq (3) in the proposal method.

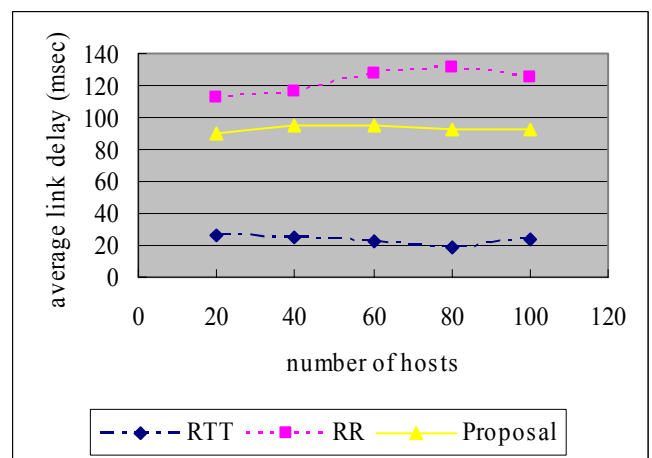


**Figure 8:** Average throughput comparison (1) effectiveness of the proposed in-degree restriction

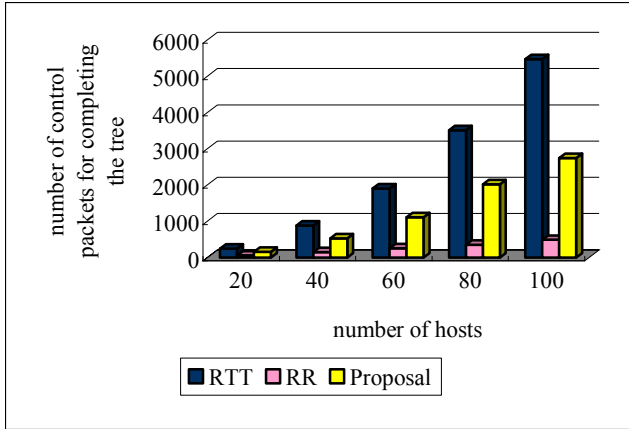
Figure 8 shows effectiveness of the in-degree restriction against no restriction case. When the in-degree is not restricted, total throughput decreases as the number of nodes increase. This is because the hosts which have small out-degrees are located in higher-position of the tree, and their descendants cannot receive at the rate which they desire consequently. As the number of hosts increase, this trend becomes quite noticeable. Therefore, the in-degree constraint has to be applied for large scale P2P streaming systems. Second, Figure 9 shows a result of throughput comparison among three methods; our proposal, RTT method and Round-Robin method. In our proposal, all hosts have proper parents and can receive at the rate which they desire by using the redefined degree parameters. On the contrary, other methods have the described problem of ‘improper relation’ in Figure 1, and some hosts can not be satisfied with the receiving rate. The more hosts join the tree, the less average throughput becomes because the hosts which receive higher rate are only in higher-positions of the tree. In this point, it is expected that our proposal can keep high delivery rate even if the number of hosts increase.



**Figure 9:** Average throughput comparison (2) effectiveness of our proposal against two conventional methods.

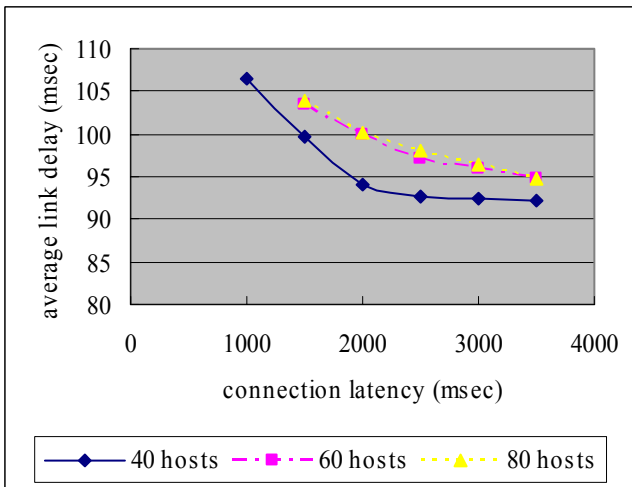


**Figure 10:** Average delay comparison among three methods.



**Figure 11:** Overhead comparison among three methods.

Third, Figure 10 and Figure 11 show that the proposal does not necessarily sacrifice the delay and overhead when compared to the other methods. The RTT method achieves minimum link delays but suffers from heavy overheads as the number of nodes increase. The RR method brings quite small overheads but its link delays are always the worst. Our proposal achieves moderate link delays and overheads, in addition to the prominent throughput advantage.



**Figure 12:** Relationship between link delay and varying connection latency in construction phase of the proposal method.

We assume that short link delay between two hosts is that they are in near position in real network or that the link is in a good condition. It is considered that, though the users are not aware of several milliseconds delay, this is an important metric from the viewpoint of using limited network resources efficiently. In the proposal method and the RTT method, a new participant host decides its parent from among the parent candidates by measuring RTTs, Figure 10 shows that the RTT method has less delay than the proposal method. This is because the new participants have all leaf hosts for the parent candidates in the RTT method. On the

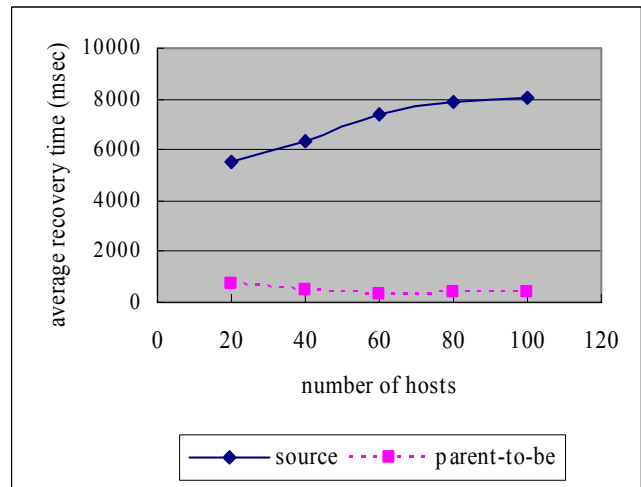
other hand, they have only the hosts which have proper degrees in the proposal method (there are less control packets in the proposal method instead). In short, the more the parent candidates are, the shorter the link delay between them becomes.

We establish a parameter called “connection latency” which limits the setup time for searching the parent candidates which have proper degrees when a new participant joins the tree. The user wants to enjoy the video or music as early as possible, and this latency is desired to be small. However, by waiting for a long time to collect more host candidates, better hosts which have large bandwidth and small link delay can be found. Therefore, there is a trade-off between the connection latency and the achieved link delay as Figure 12 shows. We prioritize the link delay because the link is used continuously. Thus, the connection latency is decided 5 seconds for which sufficient parent candidates are found in our simulation. This parameter may be changed according to the network size of the ALM tree.

#### 4.1.2 Recovery phase

Next, we evaluate the average recovery time and the overhead which affects the ALM performance against node departures. The former is the time until a new parent of the descendant is found after a host in higher position of the tree departs the ALM. The latter is the number of control packets for exchanging information with the neighbors.

As described in Section 3.3, we compared the two cases. First, all new participant hosts memorize their parent candidates as the parent-to-be which are found at Step 2, and request them to reconnect directly when their ancestors depart the tree. Second, the descendants rejoin to the source after being aware of his parent’s departure and decide a new parent as a result of Steps 1~4. We here assume the probability of host departure is 10%.

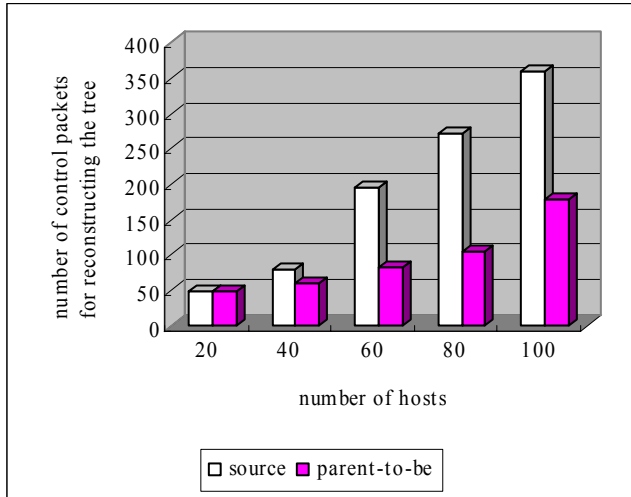


**Figure 13:** Recovery time comparison when node departure happens (departure rate is 10%).

Figure 13 shows that the “parent-to-be” method takes much less time to recover the tree. This is because it needs only the time for exchanging packets directly with the parent-to-be regardless of the network size. To the contrary, in the “source” method case, the



new participant host waits for about 5 seconds to collect sufficient number of candidates of the parent host. Although we can shorten the latency in return for decreasing the number of the parent candidates, quality of available routes decreases as described in Section 4.1. Even if we do so, however, the parent-to-be method has an advantage in terms of the recovery time.

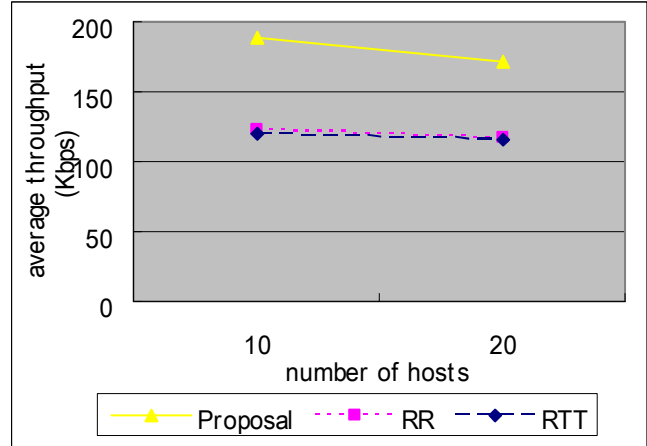


**Figure 14:** Overhead comparison between two methods in the recovery phase.

Figure 14 compares overheads of the two methods in the recovery phase. The control packets are two kinds; one is for a join request message to rejoin the tree. This is redirected to the whole tree until the new participant’s parent is decided. The other is to exchange information with the neighbors when joining and departing of hosts happen. Since the packets for host departure are common for both methods and the packets for host join are little in small trees, the total number of control packets is about the same when the tree is small. However, as the network size grows, the packets for joining process increase. Then, the parent-to-be method shows another advantage in terms of overhead.

## 4.2 Implementation results

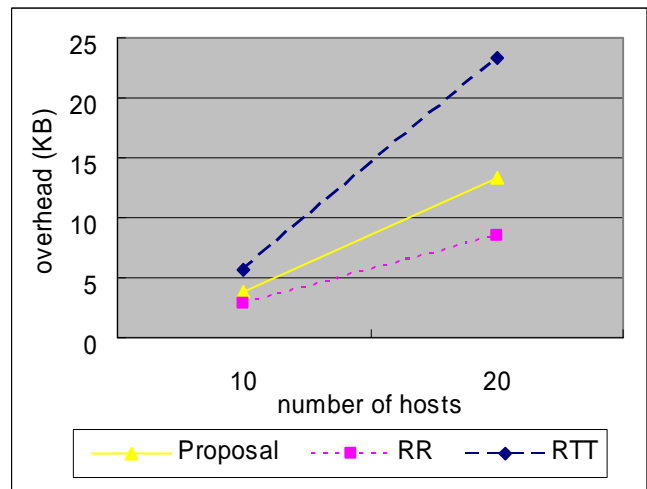
We also implemented our proposal into several computers and observed its performances over actual networks. In this experiment, video stream is encoded by H.263+ and the layering is carried out in a temporal scalable manner, by simply splitting I-pictures and P-pictures into two layers; I-pictures belong to the base layer and P-pictures belong to the second layer. The streaming rates of two layers are intentionally allocated to {125, 125} (Kbps), and the host degree is set to 1 or 2 randomly. A host whose degree is 1 can have one host which receives I-pictures. A host whose degree is 2 can have one host which receives both I-pictures and P-pictures or two hosts which receives I-pictures only. Host computers are located inside our university campus having two different locations in Japan. The number of hosts varies from 10 to 20. We also implemented two methods (RTT and Round-Robin) and compared their performances with our proposal.



**Figure 15:** Average throughput comparison among three methods in implementations.

Figure 15 shows a measured average throughput of all hosts among three methods; our proposal, RTT method and Round-Robin method. The results of the implementations are very similar to the simulation in Figure 9. In this point, we confirm that our proposal satisfies hosts’ demands on throughput.

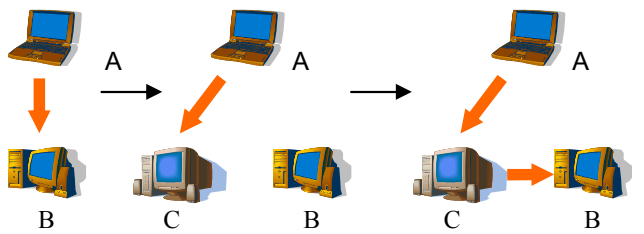
Figure 16 shows an overhead comparison result of our proposal, the RR method and the RTT method. The RR method generates lower overheads than other methods as expected from Figure 11. Our proposal generates lower overheads than the RTT method because, in our proposal, a host stops forwarding the join-request when its parent candidate has been found. From these points, we can conclude that our method actually achieves good throughput over the actual network while suppressing signaling overheads.



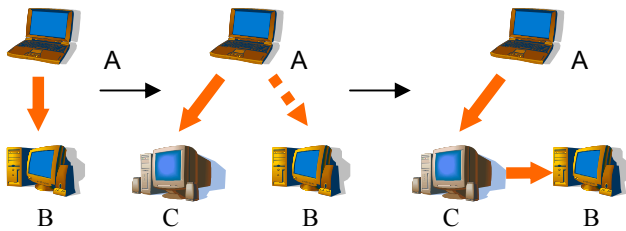
**Figure.16:** Overhead comparison among three methods in implementations.

We finally focus on the tree reconstruction process which causes interruption of content delivery during route change. In addition to the reconstruction process described in the previous section, which we call "normal" in this paper, we also consider a method of transient duplicated packet delivery to children hosts of which routes will be changed. This strategy is called "bi-casting" in this paper and is expected to reduce an interruption period to resume stream decoding.

Figure 17(a) shows an example of the normal tree reconstruction, in which new participant C sends a join request to host A having child host B. The interruption delay is caused in host B by the normal reconstruction until a new route from host C to host B is established. However, when bicasting is applied as shown in Figure 17(b), host A temporarily sends packets to both hosts B and C until host B starts to receive from host C and the interruption delay can be reduced. The length of the interruption delay mainly depends on three parameters; (a) recovery delay to establish a new route, (b) data forwarding delay over the new route through host C, and (c) delay until receiving a new I picture for intra refresh. The sum of the first two delays corresponds to the recovery time in Figure 13, which is observed in the network layer. The last delay, which is video specific and reflects the application layer, yields to a sending interval of I-pictures which can be decoded without packets of previous frames. In our experiment, since a host encodes two I-pictures per second, average delay to receive a next I-picture will be 250ms (half of 500ms).



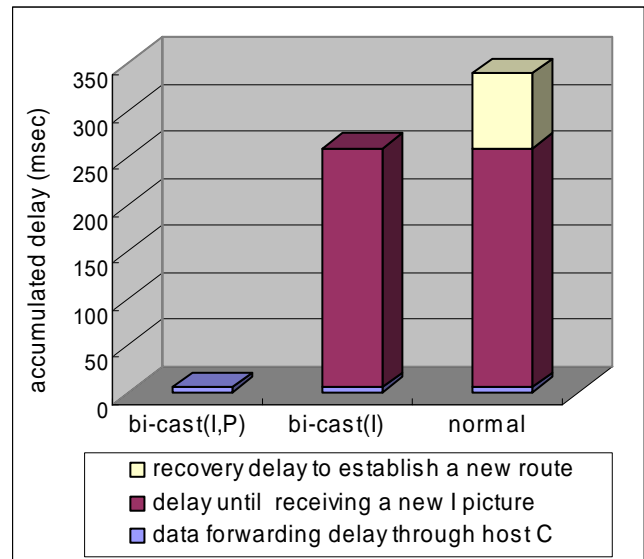
(a) Tree reconstruction without bi-casting (called "normal").



(b) Tree reconstruction with bi-casting to two hosts.

**Figure 17:** Examples of tree reconstruction with or without stream bicasting: (a) without bicasting, and (b) with bi-casting.

Figure 18 shows the delay comparison when the tree reconstruction happened. In our implementation which was deployed over rather high speed networks, observed average delays are as follows: the recovery delay for new route setup is about 80 msec, the data forwarding delay through a new host is about 5 msec, and the delay to receive a new I-picture is about 250 msec. When host A does not bi-cast ("normal" reconstruction in Figure 18), the interruption delay is provided by the sum of the three delays and amounts to about 335 msec. Then we investigate the effect of bi-casting, which can eliminate the route recovery delay at least. When two free degrees are available, host A temporarily bi-casts both I-pictures and P-pictures to host B (bi-cast (I,P) in Figure 18). When no P-pictures are lost thanks to fast response by host A, the delay to wait for a new I-picture can be also eliminated and only the data forwarding delay remains. Therefore, in this case, the interruption delay can be reduced to about 5 msec. On the other hand, when only one free degree is available, host A can not bicast both I-pictures and P-pictures. Then host A temporarily bi-casts only I-pictures to host B according to the degree constraint and the average interruption delay becomes about 255 msec which is the sum of the delay to wait for a new I-picture and the data forwarding delay. When tree reconstruction happens, it did not make us feel serious quality degradation due to bi-casting.



**Figure.18** Measurement of the average interruption delay caused by tree reconstruction; "bi-cast (I,P)": is the case that host A bi-casts both I-pictures and P-pictures, "bi-cast (I)" is the case that host A bi-casts only I-pictures, and "normal" is the case that host A does not bi-cast any pictures.

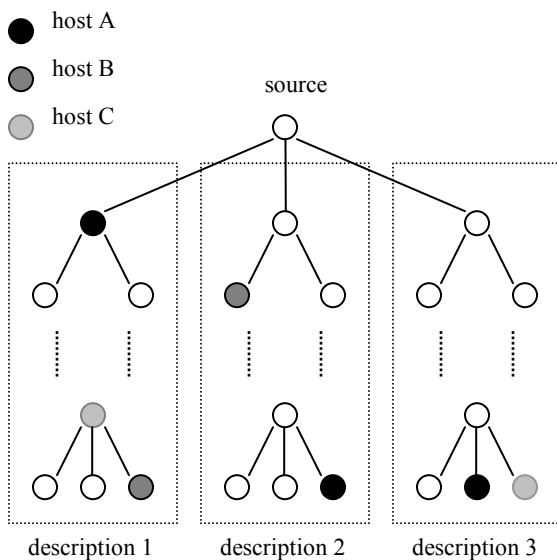
## 5. RELATED WORK

PALS (Peer-to-Peer Adaptive Layered Streaming) [7] is one of the streaming systems using layered coding. Several sending peers (SPAL) transmit the streaming data to one receiving peer (RPAL) on the Internet. This is a receiver-driven style, and the receiver determines the number of the layers dynamically based on the



throughput from its parents. In order to stabilize the decoded video quality under the condition that each SPAL cannot send data constantly, the RPAL sets the short buffering period ( $\Delta$ ) and the received packets are allocated from the base layer per this period. When the total throughput is enough (the buffering of the highest layer at the period is successful), RPAL increases the number of layers. Otherwise, it drops the highest layer. By using layered coding, the video quality can be coordinated according to the receiving condition. However, its scalability is deemed not sufficient because multiple parent hosts have one child host.

MDC (Multiple Description Coding) is the technique which encodes the data to multiple sub-streams. The MDC decoder can reproduce the data with the quality commensurate to the number of descriptions it receives, while layered coding needs the reception of the base layer, on the other hand, for every subset of description to be decodable. Then, the ALM tree which is robust against the changeable network can be constructed by using this MDC [13][17]. In this case, an ALM tree consists of multiple sub-trees corresponding to respective descriptions, each host is an interior host in one of the sub-trees, and is a leaf host simultaneously in remaining sub-trees as Figure 19 shows.



**Figure 19:** An ALM tree using MDC

By this way, when host A in a higher position in tree 1 leaves the ALM session, the descendants of the tree can not receive the description 1, but they are not influenced in other description tree because host A is their leaf host. Moreover, each host can determine the number of the description trees to which it joins according to its receiving condition. Thus, this system can be adaptive according to internet heterogeneity. However, when the descriptions are increased in order to cope with the heterogeneity, inefficient relationship between parents and children (their real distances are long) grows and system control is to be difficult.

Layered MDC [13] tries to solve this problem. By dividing each description into some layers, the host can receive the data from its neighbor hosts. In this ALM tree using layered MDC, the

constructing way of each description tree is considered layered coding. Our proposal method which this paper presents also has a relationship to the layered MDC tree, and it constructs the finished ALM tree by itself. We think that the ALM using layered MDC will be more promising by applying our method.

## 6. CONCLUSIONS AND FUTURE WORK

We described a new approach to construct ALM trees for layered video stream. By using newly defined degree parameters and constructing trees, all hosts are easily satisfied with receiving rates which they desire. When tree reconstruction happens, the time lag can be reduced by sending streams to all the children and a new host. We have investigated its validity in terms of throughput, delay and overhead. Especially, throughput which should be given priority in streaming applications is much better than that of traditional way.

As future work, it is important not only to improve the stationary throughput performance but also to provide the robustness against the dynamic behavior of the ALM trees. Introduction of robust route maintenance approaches such as [10] [12] and network-wide evaluation via implementation are now considered. Furthermore, our current proposal method constructs the tree based on the in/out-degree parameter which is determined by each host at first and is static one while the host is in the ALM network for the present. However, each host may increase or decrease its input or output depending on the network condition. Thus, we should consider a system which reconstructs the tree adapted to the dynamic network behavior.

## 7. ACKNOWLEDGMENTS

This research was supported in part by the NICT R&D project "Broadcast System Using Communication Network" and Grants-in-Aid for Scientific Research of the Ministry of Education of Japan on "Stream Caching for New Generation Content Delivery Networks and its Ubiquitous Extension."

## 8. REFERENCES

- [1] A.Legout and E.W.Biersack, "PLM: Fast Convergence for Cumulative Layered Multicast Transmission Schemes," ACM SIGMETRICS 2000.
- [2] I.Yamaguchi, K.Homma and J.Katto, "A Study on Receiver-Driven Layered Multicast with TCP-Friendly Rate Control," IEICE Tech Report, IN 2002-105 (in Japanese).
- [3] Gu-In.Kwon and John.W.Byers, "Smooth Multiple Multicast Congestion Control," IEEE INFOCOM 2003.
- [4] T.S.Eugene.Ng, Yang-hua.Chu, Sanjay.G.Rao, Kunwadee Sripanidkulchai and Hui Zhang, "Measurement-Based Optimization Techniques for Bandwidth-Demanding Peer-to-Peer Systems," IEEE INFOCOM 2003.
- [5] John.Jannotti, David.K.Gifford, Kirk.L.Johnson, M.Frans.Kaashoek and James.W.O'TooleJr, "Overcast: Reliable Multicasting with an Overlay Network," Cisco Systems, Oct.2000.

- [6] Reza Rejaie and Shad Stafford, "A Framework for Architecting Peer-to-Peer Receiver-driven Overlays," ACM NOSSDAV 2004.
- [7] Reza Rejaie and Antonio Ortega, " PALS : Peer-to-Peer Adaptive Layered Streaming," ACM NOSSDAV 2003.
- [8] Eli Brosh and Yuval Shavitt, "Approximation and Heuristic Algorithms for Minimum Delay Application-Layer Multicast Trees," IEEE INFOCOM 2004.
- [9] Duc A. Tran, Kien A. Hua and Tai T. Do, "A Peer-to-Peer Architecture for Media Streaming," IEEE JSAC Special Issue on Advances in Overlay Networks, 2003.
- [10] Mengkun Yang and Zongming Fei, "A Proactive Approach to Reconstructing Overlay Multicast Trees," IEEE INFOCOM 2004.
- [11] "ns-2" <http://www.isi.edu/nsnam/ns/>.
- [12] Y. Kunichika, T. Kusumoto, J. Katto and S. Okubo, "Application Layer Multicast with Proactive Route Maintenance over Redundant Overlay Trees," 2004 Packet Video Workshop.
- [13] V. N. Padmanabhan, H. J. Wang, and P. A. Chou. "Layered Multiple Description Coding." 2003 Packet Video Workshop.
- [14] V. N. Padmanabhan, H. J. Wang, and P. A. Chou. "Supporting Heterogeneity and Congestion Control in Peer-to-Peer Multicast Streaming," IPTPS, February 2004.
- [15] J.A. Pouwelse, P. Garbacki, D.H.J. Epema, H.J. Sips, "The Bittorrent P2P File-sharing System: Measurements and Analysis", 4th International Workshop on Peer-to-Peer Systems (IPTPS'05), Feb 2005.
- [16] Xinyan Zhang, Jiangchuan Liu, Bo Li, and Tak-Shing Peter Yum, "CoolStreaming/DONet: A Data-driven Overlay Network for Peer-to-Peer Live Media Streaming," IEEE INFOCOM 2005.
- [17] V. N. Padmanabhan, H. J. Wang, and P. A. Chou, "Distributing Streaming Media Content Using Cooperative Networking." ACM. NOSSDAV 2002.
- [18] M. Castro, P. Druschel, A-M. Kermarrec, A. Nandi, A. Rowstron and A. Singh, "SplitStream: High-bandwidth multicast in a cooperative environment", SOSP 2003.
- [19] Wei Tsang Ooi, "Dagster: Contributor Aware End-Host Multicast for Media Streaming in Heterogeneous Environment", MMCN 2005.