Implementation of Throughput Maximized ALM using Layered Video Coding

Masato OGURO Yohei OKADA Jiro KATTO Sakae OKUBO
Graduate School of Science and Engineering, Waseda University

1. INTRODUCTION

Recently, multimedia contents such as music and video have been increasing with the advent of broadband links like ADSL and FTTH. There is a limit in a simple server/client model to deliver large data to a huge number of clients because of much load on the server and network congestion around the server. In order to overcome these problems, there has been an increasing interest in application layer multicasts (ALM). ALMs do not depend on network devices such as routers and switches but rely on end hosts. All the data packets are sent as unicast packets and forwarded for multicasting from one host to another on the overlay. In addition, the multicasting can be enhanced by making each host receive at a different rate according to its environment by using layered coding.

We proposed a new approach to construct the ALM trees for layered coding [1]. Our goal is to maximize the total throughput of all hosts which are located in various environments by using layered coding and we showed the effectiveness of our proposal by simulation evaluations [1]. In this paper, we implement our proposal into several computers and evaluate the performance in a real network.

2. PROPOSED METHOD

2.1. Enhanced definition of the degree parameter

A degree parameter originally means how many child hosts a parent host can maintain. The value depends on the parent's forwarding capacity. For example, when the forwarding capacity of host *i* is *Fi* and a single streaming rate is *R*, host *i*'s degree *Di* is calculated as follows:

$$Di = Fi / R \qquad (1)$$

But in ALM using layered coding, the streaming rate *R* is not a fixed value so we should redefine the degree.

Firstly, we define *R*1 which represents the base layer streaming rate instead of *R* in (1). So we redefine degree *Di* as follows:

$$Di = Fi / R1 \qquad (2)$$

Second, *R*1 is assumed to be "1" and the accumulative rate of each layer is shown by real number as the ratio to *R*1.

As a result, a new degree parameter (*Di*) means how many base layers a parent host can have. For example, when rates of respective layer are {50,100,200} (kbps) and host *i*'s sending capacity is 500 (kbps), the degree of host *i* is 10 according to (2) and the ratio of accumulative rates is 1:3:7. Hence host *i* can accommodate 4 hosts which receive a base layer and 2 hosts which receive a base layer and a second layer. We call this degree parameter "out-degree". Clearly, this definition can also be applied to the receiving capacity. We call this degree parameter "in-degree".

Finally, we add the restriction in all hosts as follows:

$$\text{out-degree} \geqq \text{in-degree} \qquad (3)$$

This is because, if we allow a host which can send little data and receive much, other hosts receive less data. The total throughput of the tree is improved by putting this restriction.

2.2. The ALM Tree Construction Algorithm

In our proposal, we construct the ALM trees by using these redefined degree parameters. We demonstrate the process of the ALM trees construction in Figure 1. Every host maintains the degree information of itself and its children.

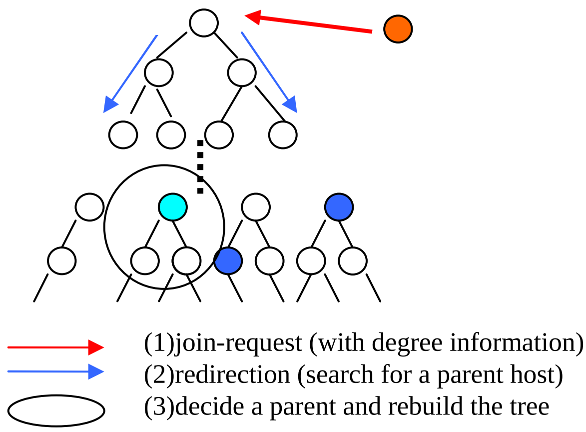

Figure.1. Process of the ALM Tree Construction

Step1.Send a join message to the source host

A new host N sends a join-request message with degree information to the source host S to participate in the tree.

Step2.Search for a parent host

The host S receiving a join-request message refers to degree parameters of itself, its children and host N, and then it checks the following two conditions:

① its（remaining）out-degree≧N's in-degree
② N's out-degree≧its children's out-degrees

At first, the host S checks the condition ① and, if it is satisfied, the host S becomes a parent of host N. Otherwise, it checks the condition ② and, if it is satisfied, the host S becomes a parent of host N. Otherwise, the host forwards a join-request message to its children and this process is repeated until a parent candidate is found.

Step3. Decision of a parent and local reconstruction of a tree

During the parent search process, host P which sends a response message first is decided to be the parent host of host N and host N joins the tree. Under the condition ① described at

Step2, host N is simply connected to host P. Under the condition ②, host P needs to rearrange connections in local area which includes host P and P's children, and host N is connected to host P. At this time, total out-degree of the local area is equal to or bigger than in-degree by (3).

3. EVALUATIONS

We implemented our proposal method into several computers and evaluated the performance of our system in a real network. In the source host, the streaming data from a capture device is encoded by H.263+ and is split into two layers (I pictures and P pictures) by temporal scalability. The streaming rates of two the layers are about {120,120} (Kbps), where the frame rate is 10(fps) and the frame interval of I pictures is 5. The number of hosts varies from 10 to 20 and the host degree is set to 1 or 2 randomly. We also implemented two methods (RTT and Round-Robin) and compared their performances with our proposal. In case of the RTT method, a host forwards the join-request repeatedly until the leaf hosts receive it and a parent decision is done by measuring their RTTs. In the RR method, a host forwards the join-request to a single child chosen in a round-robin manner until the leaf host receives it.

3.1. Throughput Comparison

Fig.2. shows a comparison of average throughputs of our proposal and the RR method and the RTT method. Our proposal is better than other methods in terms of throughput. In our system, all hosts can receive the layers which they desire by using the new degree parameter and reconstructing the tree locally. In the RR method and the RTT method, some hosts can not be satisfied with their receiving rates because the hosts which receive higher rates are located only in higher-positions of the tree. The more hosts participate in the tree, the more hosts are in lower-positions, hence the average throughput decreases.

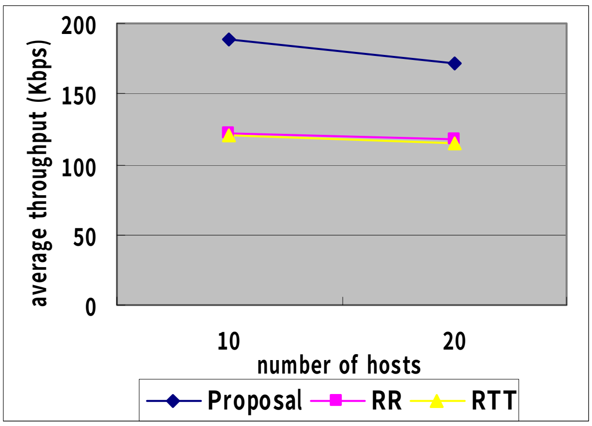

Figure.2. Comparison of average throughputs

3.2. Overhead comparison

Fig.3. shows a comparison of overheads of our proposal, the RR method and the RTT method. The RR method generates lower overheads than other systems as expected. Our proposal system 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. In this point, we can conclude that our method achieves good throughput while suppressing signaling overheads.

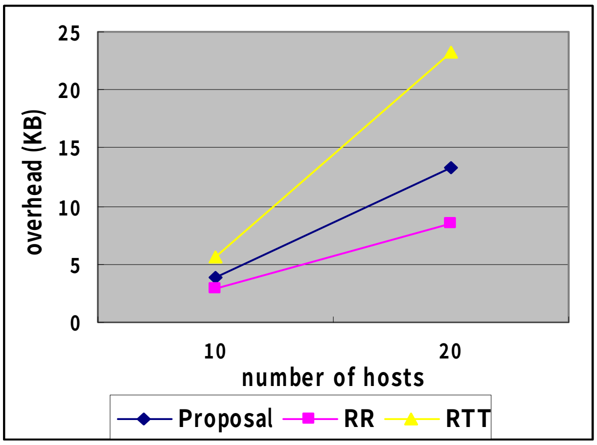

Figure.3. Comparison of overheads

3.3. Measurement of the time lag caused by tree reconstruction

Fig.4. shows a tree reconstruction example. New host C sends a join request to host A. Then host A compares B's out-degree with C's one. In this case, B's degree is 1 and C's degree is 2 hence tree reconstruction begins. The time lag is caused by normal reconstruction. But in our implementation, host A temporally sends a base layer to both host B and C until host B starts to receive data from host C and the time lag can be reduced to almost 0 sec. When tree reconstruction happens, it does not influence hosts in lower-positions.

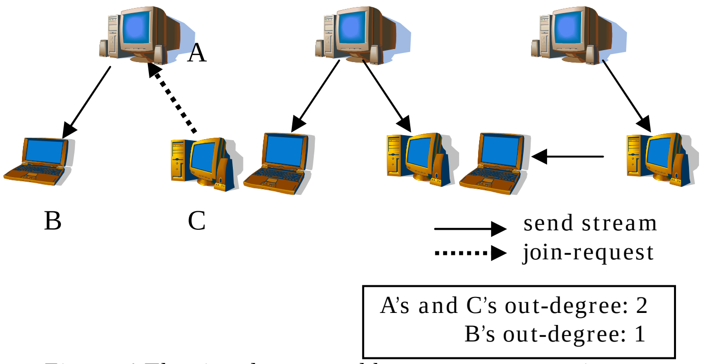

Figure.4.The time lag caused by tree reconstruction

4. CONCLUSIONS AND FUTURE WORK

We implemented throughput maximized ALM and evaluated its performance in a real network. All hosts are satisfied with receiving rates which they desire by using new degree parameters. When tree reconstruction happens, the time lag can be reduced to almost 0 sec by sending streams to all the children and a new host. As future works, we consider to provide robustness against dynamic behaviors of the ALM trees. Robust route maintenance approach such as [2] can be considered.

5. REFERENCES

[1]Y. Okada, M .Oguro, J .Katto "A New Approach for the Construction of ALM Trees using Layered Coding "IEICE Tech Report, IN2004-134, Dec.2004.

[2]Y. Kunichica, T. Kusumoto, J. Katto and S. Okubo "Application Layer Multicast with Proactive Route Maintenance over Redundant Overlay Trees" 2004 Packet Video Workshop

* This study has been carried out as part of the NICT R&D project on "Broadcasting by use of telecommunication networks."