Application Layer Traffic Engineering for Multipoint Communication System

Yumi Nakayama
Graduate school of Engineering
Nagoya University
Nagoya, Japan
yumi@ucl.nuee.nagoya-u.ac.jp

Katsuhiko Kaji
Graduate school of Engineering
Nagoya University
Nagoya, Japan
kaji@nuee.nagoya-u.ac.jp

Nobuo Kawaguchi
Graduate school of Engineering
Nagoya University
Nagoya, Japan
kawaguti@nagoya-u.jp

***Abstract*—Recent advancement of network systems enables various kinds of remote communication. In this paper, we focus on person-to-person multipoint communication. When we use multipoint communication systems, it's preferable to make maximum use of the available network resources to obtain the maximum communication quality. However, current systems are not possible to utilize the network resources effectively. In this paper, we propose a framework to describe a multipoint communication policy which enables "application layer traffic engineering" for multipoint communication systems. By using our framework, one can easily control the multipoint communication traffic like a "traffic engineering" over the broadband networks. To exemplify the effectiveness of the framework, we utilize SAMTK-3D which is a multipoint communication system in a three-dimensional virtual space. By using SAMTK-3D, we can reproduce an event held on a real space in a virtual space and enable a remote participation such that the mutual communication between a real space and a virtual space is possible. We also implement a policy management client which can visualize traffic of the communication and control the multipoint communication policy.**

***Keywords*-multipoint communication; traffic engineering; network;**

I. Introduction

Recent advancement of network systems enables various kinds of remote communication. Video streaming, video communication systems and communication services in three-dimensional virtual space also utilize network services.

There are number of studies for multipoint communication systems so far[1][2][3][4][5]. When we use multipoint communication systems, it's preferable to make maximum use of the available network resources to obtain the maximum communication quality. However, current systems and studies are not possible to utilize the network resources effectively.

In this paper, we propose a framework to describe a multipoint communication policy which enables "application layer traffic engineering" for multipoint communication systems. By using our framework, one can easily control the multipoint communication traffic like a "traffic engineering" over the broadband networks. In the real world, the network condition of clients are various. For example, some clients may have only IPv4 addresses, others may have only IPv6 addresses. IPv4 clients can't communicate directly with IPv6 clients. If there is a dual-stack client(which has both IPv4 and IPv6 addresses), it can transfer the packets over the different IP versions. However, most of current applications don't support this behavior. By using our "application layer traffic engineering" framework, we can manually control the packets transfer over the clients.

To exemplify the effectiveness of the framework, we utilize SAMTK-3D which is a multipoint communication system in a three-dimensional virtual space. By using SAMTK-3D, we can reproduce an event held on a real space in a virtual space and enable a remote participation such that the mutual communication between a real space and a virtual space is possible. We also implement a policy management client which can visualize traffic of the communication and control the multipoint communication policy. It enables to adapt communication to network resources.

In section II, we propose application layer traffic engineering. In section III, we make a proposal of multipoint communication policy. In section IV, we propose policy management client. While in section V, we describe a prototype of multipoint communication system and policy management client. In section VI we give our conclusion.

II. Application Layer Traffic Engineering

We propose "application layer traffic engineering". First, we describe traffic engineering and a related service. Then we describe application layer traffic engineering.

A. *Traffic Engineering*

Traffic engineering encompasses the application of technology and scientific principles to the measurement, characterization, modeling, and control of the Internet traffic[6]. Traffic engineering is used for network layer, not for application layer.

B. *Application-Layer Traffic Optimization(ALTO)*

There is Application-Layer Traffic Optimization(ALTO) service. That will provide applications with information to perform better-than-random initial peer selection[7]. ALTO service may take different approaches at balancing factors such as maximum bandwidth, minimum cross-domain traffic, lowest cost to the user and so on. However, multipoint communication clients communicate with all other clients.

978-0-7695-4779-4/12 $26.00 © 2012 IEEE
DOI 10.1109/NBiS.2012.100

So, ALTO service isn't suitable for multipoint communication systems.

C. Application Layer Traffic Engineering

We propose application layer traffic engineering and a framework to describe multipoint communication policy. It enables to use network resources maximum. There are some multipoint communication system. For example, Polycom[8] and Skype[9]. Both of them require some kind of central servers. To enable a multipoint communication using Polycom H.323 devices, it requires a MCU(Multipoint Control Unit). H.323 devices also require direct connection between clients and MCU. Skype requires central database for ID management and super nodes for packet forwarding. Skype nodes automatically forward the packets among the super nodes to overcome the NAT problems, however, user can't control the forwarding policy. So, sometimes it can't utilize the network efficiently. There are various kinds of network situations in the real world. For using network resources maximum, the packets should be routed ideally among the client.

Figure 1 is a typical example of divided network. The communication node n_2 and node n_4 are dual-stack. But n_1 has only IPv4, n_3 has only IPv6. n_1 and n_2 can communicate directly. n_2, n_3 and n_4 also can communicate directly. However, n_1 can't communicate with n_3 and n_4 directly. If n_2 translates n_1's packets from IPv4 to IPv6, and forward it to n_3 and n_4, n_1 can communicate with n_3 and n_4 indirectly. This kind of packet forwarding should be handled.

In this paper, we work on the technique for application layer to enable efficient routing. We propose a framework for multipoint communication policy to control the packet forwarding among the clients.

It is not easy to find out the most efficient forwarding route when all clients check conditions each other and communicate in a distributed manner. So we employ Policy Management Server(PM-Server) which gathers information for all clients and sends them to Policy Management Client(PMC). PMC visualizes communication and sends control message based on multipoint communication policy. A detailed explanation of multipoint communication policy appears in section III. And a detailed explanation of PMC appears in section IV.

There are some requirements for application layer traffic engineering.

- Denote link information includes transport and protocol(section III-A).
- Calculate all of possible forwarding routes, directly or indirectly(section III-B).
- Select forwarding route(section III-B).

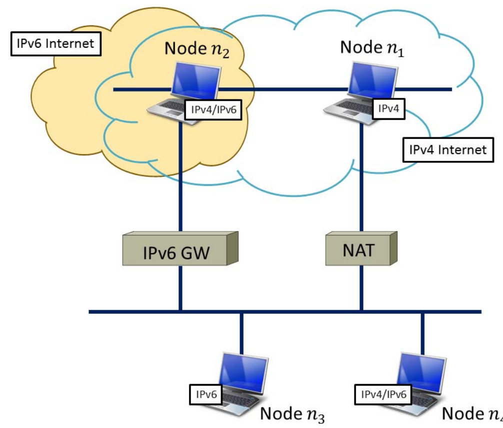

Figure 1. Example of Divided Network.

III. Multipoint Communication Policy

We propose a multipoint communication policy for multipoint communication making full use of network resources in multipoint communication system.

A. Link Description

To denote a multipoint communication policy, it is required to denote each of direct links between clients. However, it is too cumbersome to denote each of them if there are large number of clients. Instead of denoting each link, we employ group notation for describing same type of direct links among the groups. Figure 2 shows a groups and link graph of the network in Figure 1.

The followings are definitions of group notation.

- Communication node n_i means each clients.
- All nodes belong to Group *G*.
- "Group bg_i" composed of the nodes which can communicate with each other in same type of protocol.
- "Group ug_j" composed of a source node(n_s) and destination nodes(n_d). n_s can send a packet to n_d in a unidirectional link.
- The Group list *GL* includes all bg_i and ug_j.
- G_i means all groups which n_i belongs.

PM-Server knows *G* and *GL*.

B. Deciding Forwarding Route based on Multipoint Communication Policy

In our application layer multipoint communication framework, each of nodes are required to have a forwarding table for incoming packet. In the following, we will describe sending a packet from n_i to n_j. First, check that if there is a group bg_k which contains both of n_i and n_j, or check that if there is a group ug_l which contains n_i as a source node and n_j as a destination node. This means n_i can send a packet to n_j directly. If there is no group which satisfies these

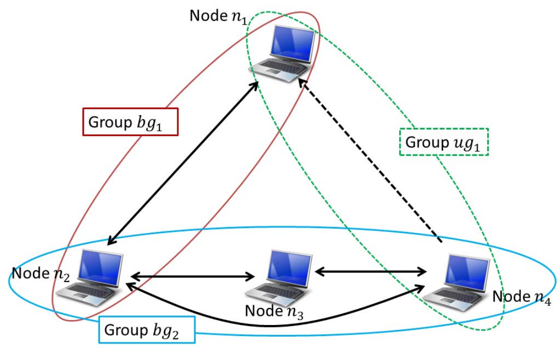

Figure 2. Example of a Typical Network Link Graph and Groups.

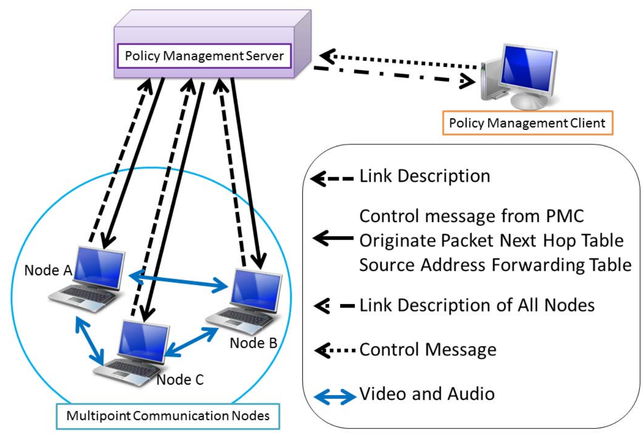

Figure 3. Flow of Information among PM-Server, PMC and Nodes

conditions, communication from n_i to n_j needs a forwarding node.

To decide a forwarding node, we employ a server which gathers link information of all nodes for simplicity. The server is named "Policy Management Server(PM-Server)". Figure 3 shows a flow of information among PM-Server, PMC and nodes. When PMC connects to PM-Server, it sends gathered link information to PMC. After PMC receives the information, PMC shows "Network Link Graph" and "Directed Graph of Groups" to a multipoint communication operator. Figure 2 is an example of "Network Link Graph". And figure 4 is an example of "Directed Graph of Groups". When PMC operator chooses two nodes, PMC shows the possible forwarding routes between the two nodes. PMC operator can select a appropriate forwarding route. After the operation is finished, PMC sends the forwarding route to PM-Server. Then PM-Server sends these routes to each node. Each node updates its own multipoint communication policy. A detailed explanation of the multipoint communication policy appears in section III-C.

In our framework, every packets contains of its source address. When nodes receive a packet, they check source address of the packet whether the source address has next hop nodes in their own "source address forwarding table". If there is next hop nodes' address in the table, the node forwards the packet to the next hops.

C. Multipoint Communication Policy Description

We denote a multipoint communication policy in the form of two tables for each node. One is "originate packet next hop table". It shows which is next hop node when the node sends its own packets to destination node. The other table is "source address forwarding table". It shows a next forwarding node based on the source address of a received packet. In the following, we describe the tables.

1) Originate Packet Next Hop Table for Node n:

- This table has two columns: Destination and Next Hop.
- Destination means a destination node which the node sends its own packet to.
- Next Hop means an adjacent node which the node sends its own packet first. Next Hop is annotated with Group to indicate the specific protocol for the link.

2) Source Address Forwarding Table for Node n:

- This table has two columns: Source and Next Hop.
- Source means a source node which the receive packet is originated.
- Next Hop means an adjacent node which the node sends packet from Source. Next Hop is annotated with Group to indicate the specific protocol for the link.

D. Example of Multipoint Communication Policy

Figure 4 shows a digraph of groups and nodes correspond to figure 2. The communication nodes n_2, n_3 and n_4 have IPv6, so they can communicate directly with each other and make group bg_2. In this situation, $n_1$and n_2 make group bg_1. And n_1 and n_4 make group ug_1. In ug_1, source node is n_4 and destination node is n_1. Table I shows bg_i including nodes and ug_1 including source node and destination node. Table II shows all forwarding routes from n_3 to n_1. And Table III shows all forwarding routes from n_4 to n_1. When PMC operator selects route 1 in table II and route 2 in table III, originate packet next hop table for n_1 and n_2 are shown in table IV and table VI respectively. And source address forwarding table for n_1 and n_2 are shown in table V and table VII. In this network, n_1 can't send to n_3 directly. According to table IV and table VII, when n_1 wants to send to n_3, n_1 sends to n_2 first. When n_2 receives packets from n_1, n_2 forwards n_1's packets to both of n_3 and n_4.

IV. Policy Management Client

For the nodes adapt their communication to their own resources, they should manage their forwarding routes. Ideally speaking, the nodes should automatically calculate optimal

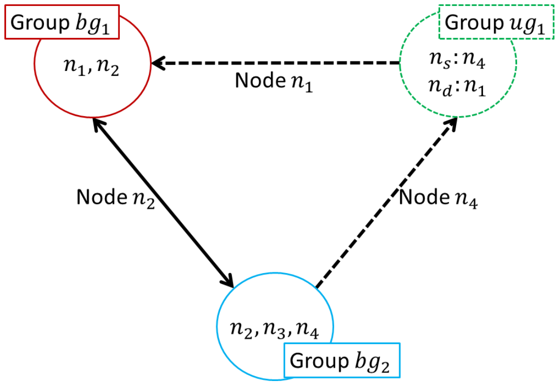

Figure 4. Directed Graph of Groups and Nodes.

Table I
GROUPS AND NODES.

Group	Nodes
bg_1	n_1, n_2
bg_2	n_2, n_3, n_4

Group	Source	Destination
ug_1	n_4	n_1

Table II
ALL ROUTES FROM n_3 TO n_1.

No.	Forwarding Routes
1	$n_3 \to n_2 \to n_1$
2	$n_3 \to n_4 \to n_1$
3	$n_3 \to n_2 \to n_4 \to n_1$
4	$n_3 \to n_4 \to n_2 \to n_1$

Table III
ALL ROUTES FROM n_4 TO n_1.

No.	Forwarding Routes
1	$n_4 \to n_1$
2	$n_4 \to n_2 \to n_1$
3	$n_4 \to n_3 \to n_2 \to n_1$

Table IV
ORIGINATE PACKET NEXT HOP TABLE FOR n_1.

Destination	Next Hop
n_1	-
n_2	$bg_1 : n_2$
n_3	$bg_1 : n_2$
n_4	$bg_1 : n_2$

Table V
SOURCE ADDRESS FORWADING TABLE FOR n_1.

Source	Next Hop
n_1	-
n_2	-
n_3	-
n_4	-

Table VI
ORIGINATE PACKET NEXT HOP TABLE FOR n_2.

Destination	Next Hop
n_1	$bg_1 : n_1$
n_2	-
n_3	$bg_2 : n_3$
n_4	$bg_2 : n_4$

Table VII
SOURCE ADDRESS FORWADING TABLE FOR n_2.

Source	Next Hop
n_1	$bg_2 : n_3, n_4$
n_2	-
n_3	$bg_1 : n_1$
n_4	$bg_1 : n_1$

forwarding routes. But it's not easy to adapt communication to network resources automatically. We propose a Policy Management Client(PMC) to enable manual operation. PMC can show all communication information to an application layer network operator. And it allow to control the forwarding route.

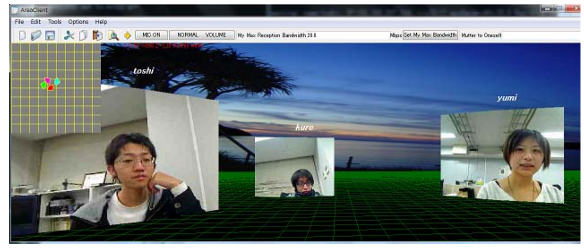

Figure 5. Screen of SAMTK-3D Client.

V. PROTOTYPE

A. Multipoint Communication System

Figure 5 shows our prototype of multipoint communication system clients' screen. We call it "SAMTK-3D". Remote participants can move in a three-dimensional virtual space and communicate with each other. The camera device captures participants' face and they can see others' video, so nonverbal communication is enabled by them.

In the following, we will describe related issues of SAMTK-3D.

1) SAMTK: We use SAMTK[10] for the implementation of SAMTK-3D. SAMTK is an abbreviation for Scalable Adaptive Multicast Tool-Kit. It is a middleware for simplifying the development of multipoint communication software[11]. SAMTK allows the developer to concentrate fully on the functionality that person wants to implement. It is implemented using the C++ Programming language. Its GUI Toolkit has been realized using Nokia's Open source Qt[12], which allows for multiplatform development. Currently, SAMTK is working on Windows, MacOS and Linux.

2) Hybrid P2P Communication: SAMTK-3D has a virtual space server and clients. The clients communicate with each other in a P2P manner. By using SAMTK for the implementation of clients, it's easy to communicate with other clients using P2P. When participants move in a three-dimensional virtual space, client will send its own direction and position to the virtual space server. The virtual space server manages and update the client's destination lists for video and audio. After that the server sends the lists to the client. The client receives destination lists, the client send audio and video streaming UDP packets to other clients. Because of using SAMTK for the implementation, the clients can use several kinds of protocol such as IPv4, IPv6, XCAST[13], Application Layer Router(ALR)[14].

3) Virtual Space Server: The virtual space server receives TCP connections from clients. It manages 3D spacial information of clients. When new client connects to the server, the server sends new client's information to other clients and all clients' information to new client. Each time clients move, the server receives clients' position and send destination

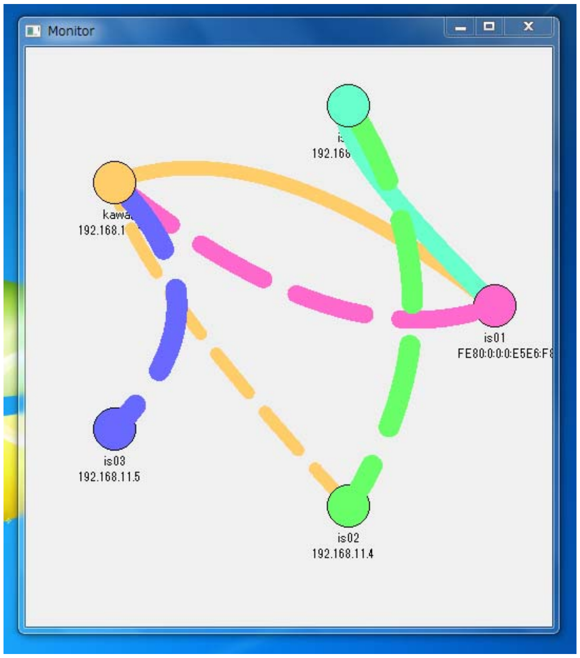

Figure 6. Screenshot of Policy Management Client.

lists including parameters to other clients. Communication between the clients and the server is only control message.

B. Policy Management Client

Figure 6 is a screenshot of the prototype of PMC for SAMTK-3D. PMC connects PM-Server, the server sends clients' link description to PMC. When the clients move or leave the virtual space, the screen is also updated. PMC sends control messages to PM-Server, then PM-Server send it to the clients. Then, the clients follow the multipoint communication policy.

The prototype of PMC also displays the status of client communication. In figure 6, PMC shows clients' name and IP address which client use. The circles mean clients, dashed lines mean IPv4 communication, other lines mean IPv6 communication. The color of communication is the same color of the source client.

C. Flow of Information

Figure 7 shows a flow of information between PM-Server, virtual space server, PMC and SAMTK-3D clients.

1) The clients register to both of virtual space server and PM-Server.
2) Virtual space server manages lists of destination of each clients.
3) Virtual space server sends lists of destination and list of all client to each client.
4) The clients send own condition(include name, IP address, port, transport and so on) and G_i.

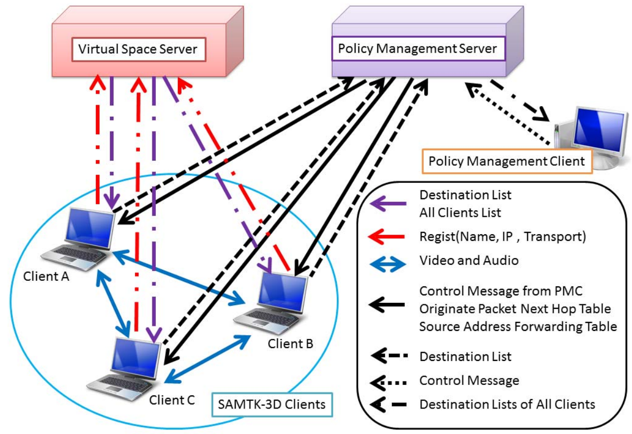

Figure 7. Flow of Information among Servers and Clients.

5) PM-Server collects the clients information and sends those to PMC.
6) PMC visualizes the nodes condition and decides communication routes based on multipoint communication policy.
7) PMC sends control messages and routes to PM-Server.
8) After PM-Server receives the messages from PMC, PM-Server sends the messages to the clients.
9) After the clients receive the message from PM-Server, the clients start to communication base on multipoint communication policy and list of destination.

When the client changes communication target, the client tells PMC through PM-Server. Then PMC updates the screen. After the operator selected the desired forwarding route, PMC sends control message to the clients through PM-Server. When clients receive the control message, clients update their multipoint communication policy. Then the clients now follow the policy selected by the operator.

VI. Conclusion

In this paper, we have proposed a framework to describe a multipoint communication policy which enables "application layer traffic engineering" for multipoint communication systems. By using our framework, one can easily control the multipoint communication traffic like a "traffic engineering" over the broadband networks.

Some of multipoint communication systems required that all nodes can communicate directly. But there are various kinds of networks in the real world. It's not easy for some of current multipoint communication systems to efficiently work in the different networks which can't communicate another directly. By using our "application layer traffic engineering" framework, we can manually control the packet forwarding routes.

We also propose a policy management client which can visualize traffic of the communication and control the communication based on the multipoint communication policy. It enables to adapt communication to network resources manually.

To exemplify the effectiveness of the framework, we utilize SAMTK-3D which is a multipoint communication system in a three-dimensional virtual space. By using SAMTK-3D, we can reproduce an event held on a real space in a virtual space and enable a remote participation such that the mutual communication between a real space and a virtual space is possible. And we also implement a prototype of policy management client for SAMTK-3D. It shows clients' name, IP address and which nodes communication with. Through using our framework, one can easily control the multipoint communication traffic.

REFERENCES

[1] C. Diot, W.Dabbous and J. Crowcroft, "Multipoint communication: a survey of protocols, functions, and mechanisms", IEEE Journal on Selected Areas in Communications, 15(3), 277?290, 1997.

[2] M. Waehlisch, T C. Schmidt and S. Venaas, "A Common API for Transparent Hybrid Multicast draft-irtf-samrg-common-api-04", Internet Draft, IETF, 2012.

[3] S. Venaas, X. Li and C. Bao, "Framework for IPv4/IPv6 Multicast Translation draft-venaas-behave-v4v6mc-framework-03.txt", Internet Draft, IETF, 2011.

[4] J. Buford and M. Kolberg, "Application Layer Multicast Extensions to RELOAD draft-samrg-sam-baseline-protocol-00", Internet Draft, IETF, 2011.

[5] C. Jennings, B. Lowekamp, S. Baset and H. Schulzrinne, "REsource LOcation And Discovery (RELOAD) Base Protocol draft-ietf-p2psip-base-20", Internet Draft, IETF, 2012.

[6] D. Awduche, A.Chiu, I. Widjaja and X. Xiao, "Overview and Principles of Internet Traffic Engineering", RFC 3272, 2002.

[7] J. Seedorf, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, 2009.

[8] Polycom, http://www.polycom.com/

[9] Skype, http://www.skype.com/

[10] SAMTK homepage, http://www.samtk.org

[11] Nobuo Kawaguchi, "SAMTK: A Toolkit for Scalable Adaptive Multicast", IETF69, SAMRG meeting materials, http://www3.ietf.org/proceedings/07jul/slides/SAMRG-2/samrg-2.ppt

[12] Qt, http://qt.nokia.com/

[13] R. Boivie, et al., "Explicit Multicast (Xcast) Concepts and Options", RFC5058, 2007.

[14] Nobuo Kawaguchi, Shuntaro Nishiura, Odira Elisha Abade, Takahiko Kurosawa, Tatsuya Jinmei, Eiichi Muramoto, "NAT Free Open Source 3D Video Conferencing using SAMTK and Application Layer Router", 2009 6th IEEE Consumer Communications & Networking Conference(CCNC2009), January 2009.