BROADCAST LEARNING SYSTEM USING MULTICAST ADDRESS

Andreas Handojo[1], Dumichen Stefano Frederik[2], Justinus Andjarwirawan[3]

Informatics Department
Petra Christian University
Surabaya, Indonesia
[1]handojo@petra.ac.id, [3]Justinus@petra.ac.id

Abstract— **Nowadays, teaching and learning process using computers becomes inseparable process, especially on information technology department. The teaching process often occurs on the laboratory, which is every student will do the learning process in front of a computer. A computer can help the learning process but also can disturb the learning process because student can be distracted to do something else. Therefore, this research will develop server and client applications to help control the process of learning and teaching that give on a computer laboratory by broadcast everything that show on computer server monitor to each client. The server can also control the keyboard, mouse, monitor and terminate or block an application that run on computer client. This application will be build using Microsoft Visual C#. Testing results showed that the number of clients does not affect the bandwidth consumption and CPU usage at broadcast. The smooth multicast depends on several factors such as type of image compression and resolution sizes. Selection resolution size and type appropriate image compression is recommended that more real-time multicast.**

Keywords- Broadcast, Learning System, Multicast, Network,

I. INTRODUCTION

Nowadays, teaching and learning process using computers becomes inseparable process, especially on information technology department. The teaching process often occurs on the laboratory, which is every student will do the learning process in front of a computer. This is very useful because that every students can have a better understanding by practicing or apply the theories they learned from class immediately. But, a computer that can help the learning process could also become something that interfere the learning process because student can be distracted to do something else. Besides that, there's another problem, usually teacher must give a practice example of what they teach using their computer and shows it using a projector in class/laboratory, but the view of projector in the class sometimes not as big and clear as needed, so student can't see it clearly.

Based on these problems, this research will develop an application that can assist a teaching and learning process by broadcast everything that show on teacher's computer monitor to each student's computer monitor, so that every students can see clearly what that the teacher doing or example that the teacher gives. This application also has a feature that can monitor what application that running on every student computer/client [3] [6] [7] and the teacher computer/server can be able to terminate it or block that

application. Another feature on this application is that computer server can be able to stop mouse [4] and keyboard function [5], so the student can pay a full attention on the lessons.

II. INTERNET MULTICAST

A normal computer communication is between a sender and a receiver. However, some applications needed to have a communication with a large number of recipients simultaneously. For example for updating a distributed database, sending the stock price to a number of brokers, and handling conference calls digitally. Internet Protocol supports multicasting process, using the class addresses D. which is for each D-class address will identifies a host group. Twenty eight bits can be used to identify the group, so that more than 250 million group can be in the same time [9] Multicast routing will be Implemented using spanning tree algorithm. Each multicast routers exchange information with their neighbors using a distance vector protocol so that each router will form a spanning tree for each group which included all the members of the group [9].

The interface addressing is divided into 4 types: unicast, multicast, broadcast and anycast [2]

- Unicast Address

The IPv4 unicast address is in the class A-C. A unicast address is uniquely identifies an interface which has an IP address. Suppose there is a computer with Ethernet card having an IP address 10.77.0.1. In the same network, there is no computer that can have the same IP address. If there is the same IP address, operating system automatically will issue a warning that there are multiple IP addresses. If a packet sent to a unicast address, that packet will be sent to the interface that have that address, so that the relation of this package delivery is one-to-one, as shows on figure 1.
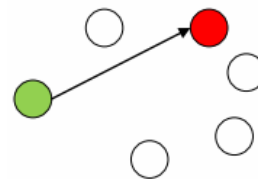

Figure 1. Delivery of Unicast address [1]

- Multicast Address

Multicast address is also available in IPv4 in class D. A multicast address identifies a group that consists of several interfaces that have IP addresses. A packet sent to a multicast

address will be processed by all members of the multicast group, as shown in Figure 2.
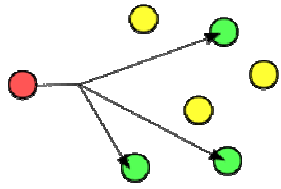

Figure 2.   Delivery of Multicast Address [2]

The advantage using this multicast address is the sender only needs to send a one-time data to all users who joined in the multicast group. So sending the multicast address can save bandwidth that needed. Suppose a user wants to stream the video data size of 1 megabytes/second. If delivery is unicast and there are 5 users, means needed 5x1 megabytes/second to send data. Besides that, at one time only one user that can receive the data. Thus, the time required to send data to all other users is n times that required for delivery to one user. This is very inefficient, because it takes time and large bandwidth

By sending data in multicast address, the user need only send data once and the data will be sent to a multicast address. Other users who want to receive this data must listen to the multicast address and a certain port number. Thus, the number of users who receive data on a multicast packet relatively does not affect the existing network.

But the weakness of the multicast address is that each packet sends using the User Datagram Protocol (UDP) packets which not guaranteed to be delivered to the recipient.

- Broadcast Address

A broadcast address is on the IPv4 only and it is the last address on a network/sub networks. Every data that sent to the broadcast address will be accepted by all the interfaces that have IP address that is on a same network/sub networks. For example, suppose there is a computer with IP address 10.77.0.1 and have a subnet mask 255.0.0.0, so the broadcast address for that network is 10,255,255,255, because it is the last address on the network. If a computer sends data to IP address 10.255.255.255, the data will be sent by all computers, ranging from 10.0.0.1 to 10,255,255,254 with subnet mask using 255.0.0.0, as shown in Figure 3
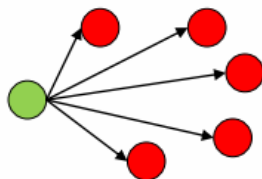

Figure 3.   Delivery in Broadcast Address [1]

- Anycast Address

Anycast address is only found in IPv6. A packet sent to an anycast address will be sent only on one device only and is usually sent to the closest device, as shown in Figure 4.
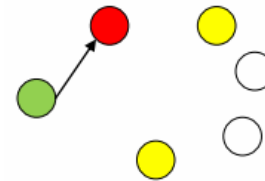

Figure 4.   Delivery on Anycast Addresses [2]

III.   DESIGN AND IMPLEMENATION

In general, the application is divided into server and client application. The server will take full control over the client, so client can give a full attention to teaching and learning process. Client only receives the command from the server and run the command. At the beginning of the system is run, the server need to create class that will be executed. After making class, the server will wait for clients to join the class. After that the server can be execute the class (figure 5).
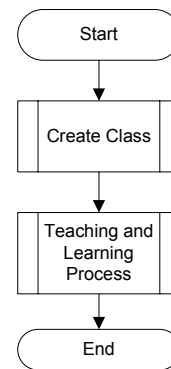

Figure 5.   Flowchart Process System

To create a class, computer server need to enter the class name and then system will send all class list (class names and the IP server) names to all existing clients in the current multicast. Then client can choose which class they want to join. As seen on figure 6.
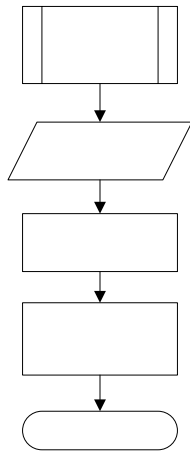
Figure 6. Flowchart Create a Class

In the teaching and learning process execution, the server will receive IP from the entire clients that enroll in the class. After that, the server can start the multicast process which is will sent everything that can be seen from computer server monitor to the client monitor. The data that will be sent to the client is a capture from computer server monitor using an image data. Besides that, server could use the multicast feature (such as view client and list the process that active on the client, block a particular application on client, disable keyboard and mouse, block USB port on client).
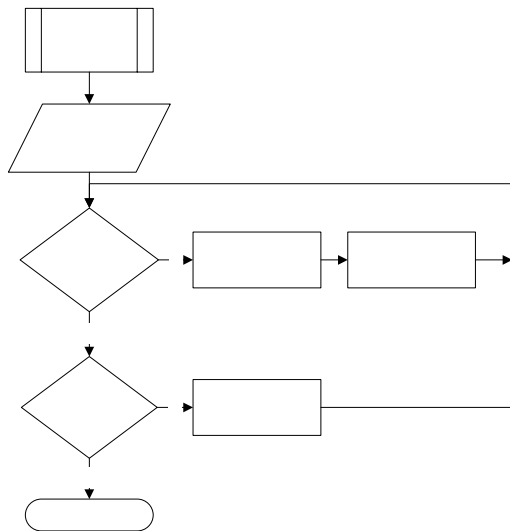

Figure 7. Flowchart Teaching Implementation Process

The implementation of the application to create a class can be seen on figure 8, user must first entry Class Name, Teacher, Day - Clock, and ports that want to use. After filling in this form, then the user is allowed to create the class.

Figure 8. Create Class Menu

The client will receive list of all class that have been created by the server (figure 9). Then the client can select the class that they want to join in.

Figure 9. Display the Class to the Client

On the server, all clients that already join in a class can be seen as figure 10. Then the server can start the teaching and learning process by start the multicast or another feature on the main control form.

Figure 10. MainControl Form

During the multicast, all clients who already join in a class will get the same view of computer server monitor, as can be seen on figure 11and 12.

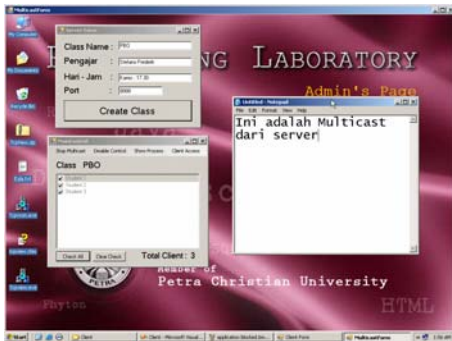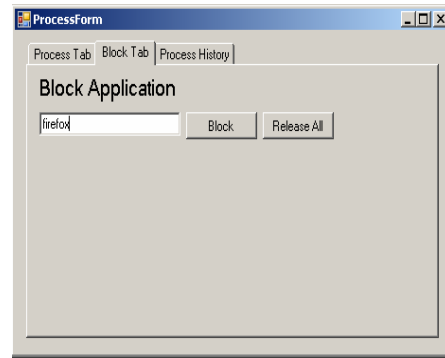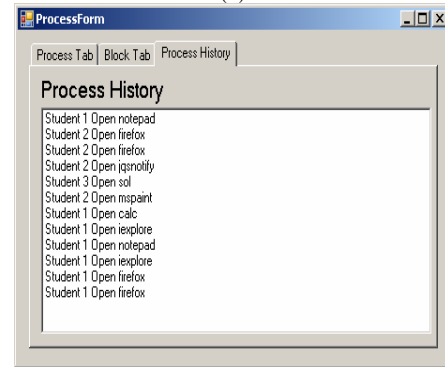Figure 11. Display on the Computer Server Monitor

Figure 12. Display on the Computer Client Monitor

Server can view all the process that currently running on the selected client (figure 13 a) or a history of application that have been use by client (figure 13 c), and if necessary server can block or kill the process that running on the client (figure 13 b).

(a)

(b)

(c)

Figure 13. (a) Client Process List Form, (b) Block Client Process Form, (c) Client Process History List Form

IV. EVALUATION

To testing the performance (especially on delay that occurs) of the system that built on this research, there's a several variable testing that will be carry out. To test the multicast delay base on type of image formats (gif, jpeg, bmp, and png) that been send from server to the client (table 1). The results show us that it appears that the biggest delay is on BMP format and the smallest delay is on GIF format.

TABLE I. DELAY TESTING BASED ON FORMAT TYPE

Format	GIF	JPEG	BMP	PNG
Experiment 1	00.88	01.48	11.22	00.90
Experiment 2	01.17	01.20	11.53	01.39
Experiment 3	01.18	00.98	11.86	01.22
Experiment 4	00.90	01.27	10.98	01.30
Experiment 5	01.18	01.01	09.29	00.95
Experiment 6	00.88	01.22	11.96	01.19
Experiment 7	01.18	01.28	12.44	00.92
Experiment 8	00.85	01.20	09.16	01.20
Experiment 9	00.94	01.18	11.85	01.05
Experiment 10	01.01	00.97	12.38	00.85
Average:	1.017	1.179	11.267	1.097

This research also conduct testing the effect of computer server monitor resolution by changing the size of the server screen resolution to 800x600, 1024x768, 1152x864, 1280x768 and 1440x900 (Table 2), but the client screen
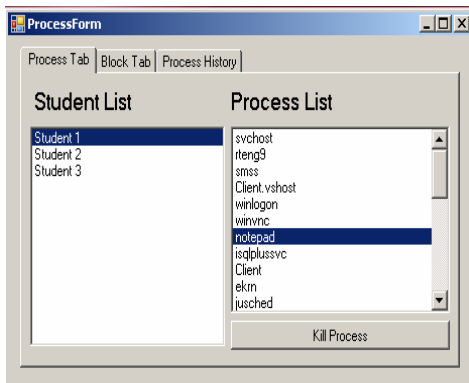
resolution used is 800 x 600. The result of this testing is that the bigger screen size that used on server the longer the delay. The same conclusion also obtained in previous research on video conference broadcast [1].

TABLE II.   DELAY TESTING BASED ON THE RESOLUTION MONITOR

Screen Resolution	800 x 600	1024 x 768	1152 x 864	1280 x 768	1440 x 900
Experiment 1	00.64	00.88	01.02	01.20	01.33
Experiment 2	00.65	00.72	00.97	01.01	01.19
Experiment 3	00.65	00.91	00.95	01.17	01.20
Experiment 4	00.69	00.82	01.18	00.99	01.24
Experiment 5	00.70	00.96	01.25	00.85	01.24
Experiment 6	00.59	00.79	01.18	01.18	01.31
Experiment 7	00.63	00.74	00.94	01.19	01.47
Experiment 8	00.59	00.49	01.12	01.18	01.32
Experiment 9	00.65	00.72	01.20	00.95	01.39
Experiment 10	00.64	00.76	00.86	01.16	00.99
Average :	0.643	0.779	1.067	1.088	1.268

The result from the addition the number of client shows that the number of client relatively do not affect the used of client's bandwidth. This test conducted on each of the client within approximately 5 minutes and tested on 5, 10, 20, 30, and 40 clients. Bandwidth usage before the multicast is about 0.01% to 0.02%. During multicast, the server sends an image data continuously so that there is an increase in the amount of bandwidth that is about 2% to 5% which means about 2 Mb to 5 Mb.
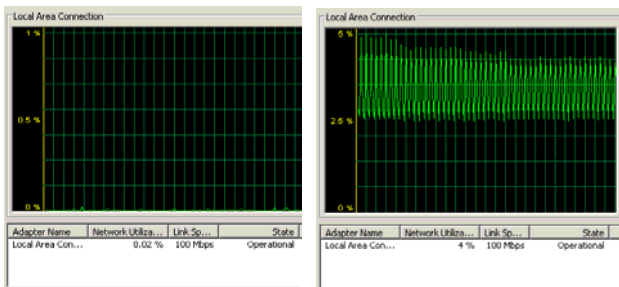

Figure 14. (a) Before Multicast (b) After Multicast for 40 client

V.   CONCLUSION

- Changes in the number of clients (with testing on 5 to 40 clients) did not have a significant affect on the CPU usage or create delay/bandwidth consuming in the event of multicast.
- Changes the number of clients also did not affect the level of success in running the feature (such as block mouse, keyboard hot key block and block significant application)
- Types of image formats that being transfer affects bandwidth consumption on multicast. BMP format type's need largest bandwidth consumption compared with other types of format (JPEG, PNG and GIF) which is affect the transfer data delay. The selection of the right type of format is strongly recommended to reduce network bandwidth consumption levels.
- Computer server screen resolution affects the transfer delay. The greater computer server screen resolution the greater delay that happened.
- On the multicast transfer data, it is possible that the received image data corrupted. This is because multicast is connectionless (using UDP), so there could be a lost package during delivery.

REFERENCES:

[1] A. Handojo, R. Chandra, and J. Andjarwirawan (2009). Video Conference Application Operate In Ipv4 Ipv6. Proc. Seminar Nasional Aplikasi Teknologi Informasi (SNTI 2009).

[2] Makofske, D and Almeroth, K. (2003). Multicast sockets practical guide for programmers. Morgan Kaufmann Publisher.

[3] Microsoft Developer Network. (2009). About hooks. Retrieved September 25, 2009, from http://msdn.microsoft.com/en-us/library/ms644959.aspx.

[4] Microsoft Developer Network. (2009). LowLevelMouseProc function. Retrieved September 25, 2009, from http://msdn.microsoft.com/en-us/library/ms644986(VS.85).aspx.

[5] Microsoft Developer Network. (2009). LowLevelKeyboardProc function. September 25, 2009, from http://msdn.microsoft.com/en-us/library/ms644985(VS.85).aspx.

[6] Microsoft Developer Network. (2009). CURSORINFO structure. Retrieved September 25, 2009, from http://msdn.microsoft.com/en-us/library/ms648381(VS.85).aspx.

[7] Microsoft Developer Network. (2009). ICONINFO structure. Retrieved September 25, 2009, from http://msdn.microsoft.com/en-us/library/ms648052(VS.85).aspx.

[8] W. Stallings. (2006. Data and Computer Communication. Upper Saddle River.

[9] A.S.Tanenbaum, (2003). Computer Networks 4th. Upper Saddle River.