PA13 OpenGL Over IP Multicasting

Lei WANG† ,Hiroei IMAI† ,Masahiro TSUNOYAMA†† , Ikuo ISHII†
Niigata University Graduate School of Science and Technology†
Faculty of Engineering, Niigata University††

1. Introduction

Networked virtual environments (net-VEs) allow multiple users to share the same object in real-time even thought those users may be located around the world. In the net-VEs, there are a lot of applications, one is allowing multiple users to interact in real-time as a player(for example, the football game) and another is permitting multiple users taking part in the net-VEs just as audience. In order to achieve the virtual environment, here is more than one way which can be taken. VRML, which is used popularly, is the method that has to download all data of virtual environment for showing the object (DIS-Java-VRML [1], for instance) But in this paper, our research try to find different way to carry out. This research examine an approach for sent the OpenGL command for redisplay by the receiver with reliable multicast communication at Intranet-based virtual environments. OpenGL is be widely used to be a software interface to graphics hardware. With OpenGL, This research can control computer-graphics technology to produce realistic pictures or ones that depart from reality in imaginative ways.

IP Multicast is a facility whereby a source can send data to multiple destinations without making unnecessary copies. Like unicast IP, IP multicast is inherently unreliable. The current multicast paradigm allows any host to create/join a multicast "group"; a packet sent to this group reaches every host that "subscribes" to it. Providing a reliable data delivery service over IP multicast is an active area of research.

This work, which consists of two technical(OpenGL and IP Multicasting), is divided two parts. On is OpenGL over Multicast library and another is client program for audience, provide another solution in the area.

2. Compose and Platform of The System

This system instrumented a Windows NT Workstation 4.0 on a 300MHz Pentium Ⅱ PC as clients and a Windows NT Workstation on a 650MHz Pentium ⅢPC as the server. At the side of client platform can run one more program as the receiver at the same time. Figure 1 shows the physical architecture for server-client net-VE on a LAN.

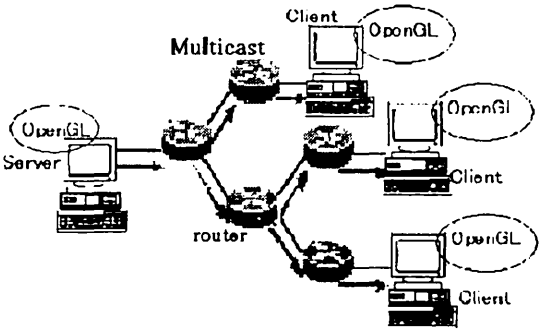

Figure-1 Physical architecture

3. Software Systems Architecture

In this section shows processing architecture for our software systems. Our software systems consist of two main parts, the first part is the library of OpenGL over the multicast, and second part is client program for graphics. The explanation below is showed in detail.

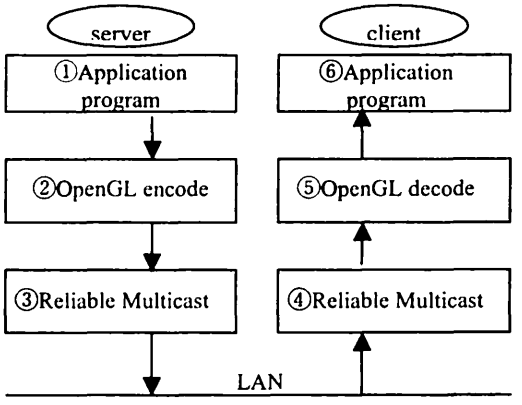

Figure-2 Software architecture

3.1 OpenGL over the Multicast library

In this part of software, the application program perform the command of OpenGL ①and at the time get the function or command of OpenGL, and then encode the same command to be transported to the receivers by

IP Multicast②③, so that at the side of the receivers display the same graphics which the server send to the receivers, for doing this, the server encode the command of OpenGL at special protocol (Figure-3) and let the received understand and decode it.

3.2 Client program

In this part of software, which perform at the receiver, the receiver display the object that formed by the encoded received command of OpenGL and decoded the message④⑤, and also obtain a windows a graphics display into which either text or 3D graphics may be drawn. In this section, the program provides a means for obtaining user input form such devices as keyboard and mice ⑥so that viewpoint can be changed, and also provides access to rendering operations, by effects of OpenGL commands on the frame buffer are controlled by this system. By this way, too many users can share the same object on the display by IP Multicast.

3.3 Encode and Decode of OpenGL command

Because the receiver can not perform the function of OpenGL even though we find out a way to transform, so we have to encode the command of OpenGL before it be sent, and the receiver decode the command and perform the function which consist with the server. We use the packet unit just like below.

a	b	c_1	----------	c_i

a: the group the commands
b: the serial number of the commands
$c_1..c_i$: the parameter

Figure 3 Encode and decode

4. Reliable Multicast

To take advantage of the bus-based nature of most LANs, we can use a hardware broadcast to send the data to the group, and then use a point-to-point acknowledgment scheme to ensure the reliability of the multicast. This is very efficient way to do multicasts, especially since it let's us take advantage of the simple bus nature of networks such as Ethernet. The protocol ｱ1[2] exhibits the following behavior:

(1) the server sends all original transmissions on a multicast address A_{org}.

(2) when required, the server retransmits a packet with sequence number i on multicast address Ai where i=0,1,2,...

(3) whenever a receiver detects a lost packet i , it transmits a NAK to the server over a point-to-point channel and subscribes to the multicast address Ai and starts a timer.

(4) the expiration of a timer without prior reception of the corresponding packet serves as the detection of a lost NAK packet, a NAK is retransmitted for the associated packet and a timer again started.

(5) on receiving packets i on Ai a receiver unsubscribes to Ai.

5. Example

At first, we can get the command of the OpenGL by which the object display at the server, and we send the encode command to the receiver by IP Multicasting protocol, the client decode the command and obtain the image both at the server and clients just like below.

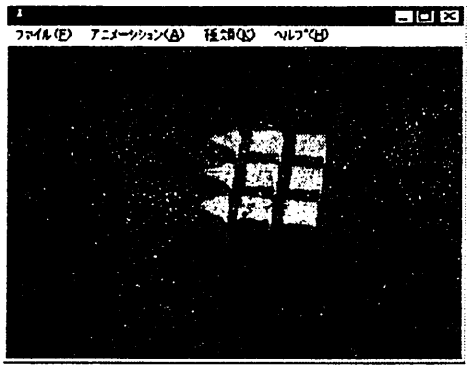

Figure 4 The output of the receivers

6. Conclusion

Up to now, we just finished the application program, which can send the simple command of OpenGL and some useful information about the object to be sent, which can be recognized by the client and redisplay the same one at the receiver, From now on, we also try to find a way to examine internet-based virtual environments, and looked at Internet-based sharing, In the age of Internet, the internet-based sharing environments are widely used in some different areas.

References

[1] Sandeep singhal, Michael Zyda: 「Networked Virtual environments」 ACM Press.SIGGRAPH Series. p251-271 1999

[2] Sneha K.Kasera,Jim Kurose and Don Towsley: 「Scalable Reliable Multicast Using Multiple Multicast Groups」 CMPSCI Technical Report TR96-73 October 1996

[3] Dave Kosiur: 「IP Multicasting The Complete guide to Interactive Networks」 Wiley computer Publishing P164-167 1998