Unicast-Multicast Bridging for CDEEP EDUSAT Satellite Network

Thesis submitted
in partial fulfillment of the
requirements for the Degree of
Master of Technology

Mohammed Nazeem Vilakkini
Roll No: 08305038
under the guidance of
Prof. Purushottam Kulkarni
and the co-guidance of
Prof. Sridhar Iyer

Department of Computer Science and Engineering,
Indian Institute of Technology, Bombay

Dissertation Approval Certificate

Department of Computer Science and Engineering

Indian Institute of Technology, Bombay

The dissertation entitled **"Unicast-Multicast Bridging for CDEEP EDUSAT Satellite Network"**, submitted by **Mohammed Nazeem Vilakkini** (Roll No: **08305038**) is approved for the degree of **Master of Technology** in **Computer Science and Engineering** from **Indian Institute of Technology, Bombay**.

PSKulkarni

Prof. Purushottam Kulkarni
CSE, IIT Bombay
Supervisor

Sridhar

Prof. Sridhar Iyer
CSE, IIT Bombay
Co-Supervisor

Prof. Bhaskaran Raman
CSE, IIT Bombay
Internal Examiner

Dr. Vijay Raisinghani
HoD (IT), NMIMS College
External Examiner

Prof. Abhay Karandikar
EE, IIT Bombay
Chairperson

Place: IIT Bombay, Mumbai
Date: 28th **June, 2010**

Declaration

I declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Mohammed Nazeem Vilakkini
Roll no: 08305038

Place: IIT Bombay, Mumbai
Date: 28^{th} **June, 2010**

Acknowledgements

I express my sincere gratitude towards my guide Prof. Prof. Purushottam Kulkarni and Co-guide Prof. Sridhar Iyer for their constant support, encouragement and inspiration throughout the project. I also thank Rahul Deshmukh, Web coordinator CDEEP, for his valuable support in helping me understand the CDEEP EDUSAT network setup and the problem definition. Also extend my gratitude to Mr. Shivraj, Technical staff of CDEEP for showing me the network setup and giving me his suggestions. I would like to thank Mr. Sachin, CDEEP staff Seminar Hall, for cooperating with me to conduct the the experiments.

Abstract

Satellite communication networks today are used for educational purposes. These networks can become vital if they can be used to reach out to the remote areas of our country. Institutions interested in joining the educational programmes through the satellite must setup a satellite transceiver. The satellite network setup is available only to place of the setup. So the courses which are to be transmitted over the satellite have to be captured live where the satellite network is available. The courses being conducted in other blocks or departments, even though they are connected by the campus network, cannot be transmitted through the satellite as they are two separate networks. So professors at other department have to come down to where the satellite link is available and give their lecture. We discuss the ways to design and implement a solution to interconnect the satellite network and the campus network so that courses conducted anywhere in the network can be transmitted through the satellite. One of the main concerns here is that the satellite communication networks are multicast whereas the campus networks may not be multicast aware. Because of this campus network cannot be directly plugged into satellite network as some kind of protocol conversion might be required.

An extensive study of interconnecting the satellite network and the campus network is made. Here we specifically target EDUSAT satellite network used by CDEEP (Centre for Distance Engineering Education Programme) and the campus network at IIT Bombay. The aim is to find ways to channel the data transmitted and received on the satellite network through the campus network with ease and making the least changes to the existing setup. We finally figure out that the tunneling method is found to be highly reliable and most easy to setup for CDEEP. This method was tested on live EDUSAT satellite network and quantitative analysis on performance of tunneling is also made.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Satellite communication networks today are used for educational purposes. An example for this is the EDUSAT satellite launched by ISRO (Indian Space Research Organisation) which made ViCTERS (Virtual Class Technology on EDUSAT for Rural Schools). It is Indias first broadband network on EDUSAT for schools. It has revolutionized classrooms through interactive IP-based technology. Apart from interactive classrooms, non-interactive channel in DTH (Direct-to-Home) in which classes are just transmitted also benefit many students. These networks become more important if they can be used to reach out knowledge to the remote areas of our country. The distance education programmes try to take the advantage of this technology to make knowledge reach even to the most rural parts of the country. The education programmes make use of these most widely deployed networks for the transmission of media that can help the courses that are given by eminent professors at reputed institutes like the Indian Institute of Technology to reach out to remote areas. The initial cost is in fact more but the benefit of this for the remote areas justifies the cost. The cost of setting up a receiving centre alone is about 3.6 Lakh according to a newsletter published by CDEEP (Centre for Distance Engineering Education Programme), IIT Bombay called ReachOut in July 2008. One of the concerns of such programmes is that there usually just one link to the satellite as setting up them is very expensive. So the courses which are to be transmitted over the satellite have to be captured live where the satellite network is available. The courses being conducted in other blocks or departments which are very well connected by the campus network cannot be transmitted through the satellite. So professors at other department have to come down to where the satellite link is available and give their lecture.

1.1 Interconnecting Campus Network and Satellite network

Our goal here is to design and implement a solution to interconnect the satellite network and the campus network so that courses conducted anywhere in the network can be transmitted through the satellite. As shown in the Figure 1.1 we can see that the campus network and satellite network are completely isolated. Terminal A and B in figure are in the satellite network. Our aim is to connect a Terminal C in campus network, as shown in Figure 1.1, to be able to connect to the satellite network.

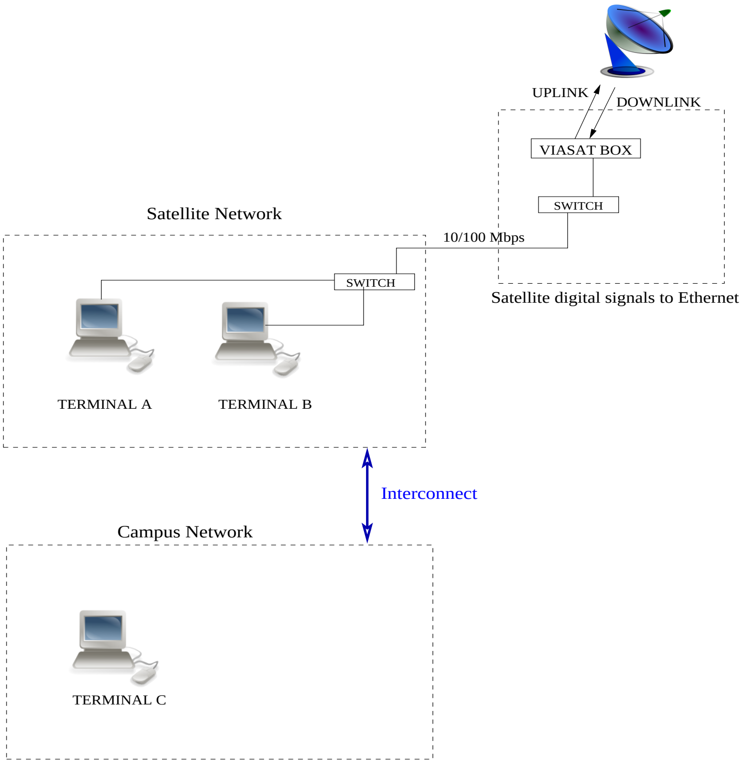


Figure 1.1: Campus network and satellite network.

One of the main concerns here is that the satellite communications networks are multicast where the campus networks may not be multicast aware. So our goal will also be to send multicast traffic through the unicast network.

1.2 Requirements

To this already existing network setup of the CDEEP ISRO collaboration what we need is a cheap and easy solution to be able to conduct lectures anywhere in the campus and using the existing network setup available inside the campus to transmit the courses through the satellite. This would be a great benefit for the Professors at their respective department as they can give their lectures at their department itself. Also, the benefit to the CDEEP is that it will be easier to schedule course transmissions as there will be more lecture halls available without increasing the load on any of the specific halls. This is not possible when there is only one class (Seminar

Hall KReSIT) where classes for live transmission on satellite can be conducted. This is because the EDUSAT satellite network is available only in the KReSIT and as of now, it is isolated from the IIT-B network. The challenge is that we cannot simply connect two networks as they are two different private networks one of which is a multicast network and other being a unicast network.

1.3 Problem Definition

Currently, the distance education programme of IIT Bombay uses the EDUSAT satellite network to transmit its live interactive courses which is available in the seminar hall of the KReSIT building. This live transmission on satellite network can only be done in this hall. If any professor wishes that his course be transmitted this way then he has to come all the way down to this building in order to give his lecture. We need to make it possible for this video facility to be setup anywhere inside the campus so that any class room can be later made into a live interactive class room. Bi-directional communication must be possible from any class room. By bi-directional we mean that in one direction the teacher must be able to deliver his lecture to the remote centers and in the other direction the students must be able to ask questions to the teacher again through the satellite network from other departments other than the KreSIT department also. When the remote centers (discussed later) have to interact with the professor giving the lecture then their audio and video are transmitted back to the lecture hall inside the campus. So we must also be able to forward the data coming from the EDUSAT network which includes audio, video and also information such as how number of remote centres that are currently participating in the course, etc to be forwarded to other departments through the IIT-B network. The goal is to make the underlying network transparent to the applications that transmit and receive transmission of courses. In addition to this the EDUSAT network has a multicast backbone and CDEEP uses multicast transmission for its courses. The IIT-B network is unicast network. So interconnecting two such types of networks makes it more challenging as some kind of protocol conversion might also be required.

One of the bottleneck here is that we must do as much small changes to the existing system. This is because CDEEP buys all equipments including both software and hardware from ISRO. So they have already setup for transmitting and receiving from one class room in KReSIT. If CDEEP desperately needs more classrooms for transmission and reception through satellite network then it can request a completely new set of hardware and software for doing this which would be expensive. So our aim is to use existing hardware in the campus network as well as in the EDUSAT satellite network. Apart from making no changes to the existing IIT-B network, the softwares and applications that are being used for the transmissions and receptions are also not be changed. This is because the CDEEP staff are trained and are used to using those which are already installed.So it will be time consuming and unproductive to change the software. So this is why our interest will particularly be at network layer though we shall compare with other possibilities as well. We may lay new network cables to extend the EDUSAT network, lay audio/video cables, use the existing IIT-B network infrastructure for the solution of the problem. Since the packets that come from the EDUSAT satellite network are multicast network, it would be a challenge for the class rooms in the unicast IIT-B network to receive these packets. This is because a simple packet forwarding will not make things work. The routers in the IIT-B network will not be able to forward the multicast packets as they are not multicast aware. If we were to direct the multicast stream to these routers, it would simply get dropped. So some kind of protocol conversion might be required to still forward the multicast stream through these routers.

Chapter 2

CDEEP Network Setup

The communication satellite used by the CDEEP is the EDUSAT. This satellite was launched with the major aim of aiding the education of remote areas of the country. The transmission of video lecture courses takes place at IIT-Bombay and there are over 72 centres across the country which takes advantage of this educational programme. The satellite network setup of EDUSAT for the CDEEP is in the KReSIT (Kanwal Rekhi School of Information Technology) building of IITB. The EDUSAT Satellite setup and the applications used for this purpose are provided by the ISRO (Indian Space Research Organization).

As shown in Figure 2.1, the satellite dish for reception and transmission is on top of the KReSIT building. The download link and upload link from the satellite are connected to a LinkStar ViaSat box. This box has a 10/100 Base T output which forms the Ethernet link to the EDUSAT network. The ViaSat box is kept at the CDEEP office on 4th floor of KReSIT (Kanwal Rekhi School of Information Technology) building inside the IIT Bombay campus. An 8-port D-Link switch is used to distribute the connection to the Seminar Hall which is in the 3rd Floor KReSIT building. There is another 8-port 3Com switch in Seminar Hall which distributes the network to the terminals (or computers with specific applications) used for transmission and reception.

The Teacher's Terminal is for the teacher so that interaction with the remote centers is possible. It can monitor which all remote centers are currently viewing the course. If any center has requested for asking a question, it gets notified on this terminal application. The video transmitted by the students terminal at the remote centers can also be received in this terminal. The Video Server is application, installed on one of the terminals, that transmits the live audio/video to the multicast server in the EDUSAT satellite network.

Very recently, a completely new satellite dish antenna and computer were setup in the CDEEP office on 4th Floor of KReSIT building for setting up the Students

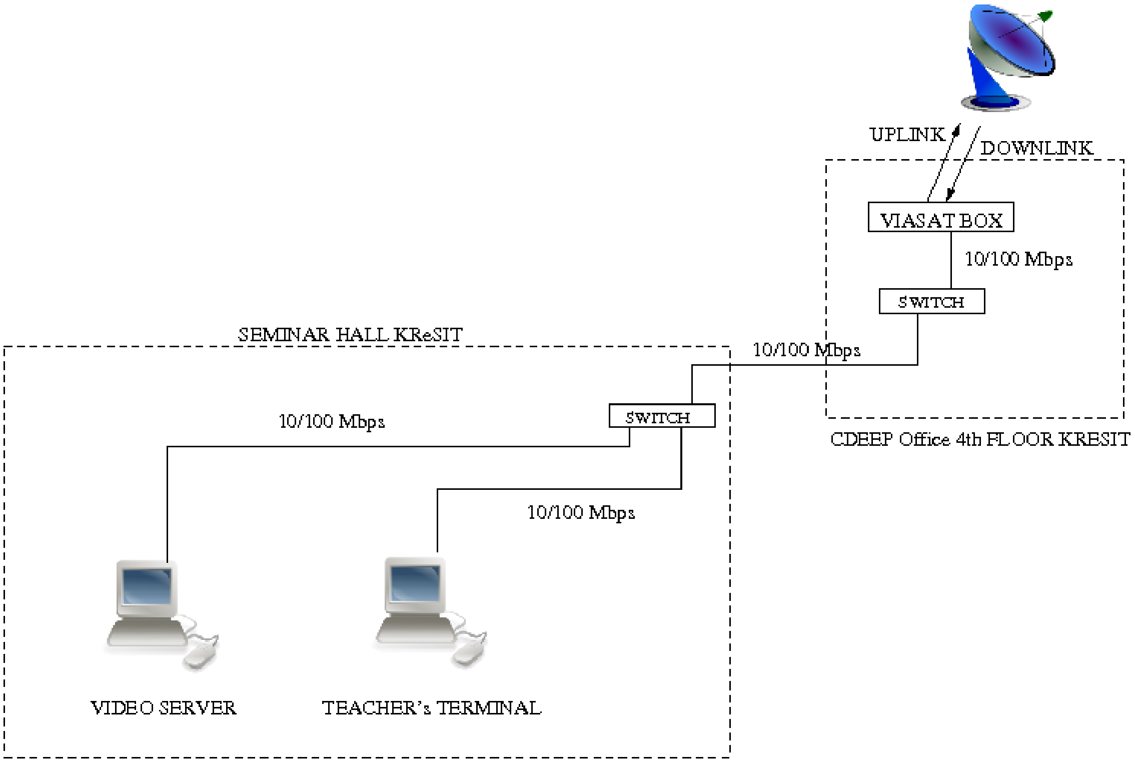


Figure 2.1: CDEEP Network setup in KReSIT.

terminal.

Currently the recording of courses takes place at 4 different places inside the campus. They are Video Lab and A1/A2 (both of which are in Mathematics Department), EEG 401 (Dept. of Electrical Engg.) and Seminar Hall 3rd Floor KReSIT Building. Out of these 4 places, the transmission of courses through the satellite takes place only at the Seminar Hall in KReSIT as the satellite dish antenna and the related setup are only available there. Whereas at the other 3 places only webcasting is done, where the transmission of courses are done through the Internet with the help of a web server. The web transmission of courses done by CDEEP is independent of satellite transmission and is also done live. The main difference between satellite transmission and web transmission is that satellite courses are interactive, that is, the students can interact live and ask questions to the teacher but in web transmissions the students can only view the course.

As shown in Figure 2.2, the transmission of courses is received by Remote Centres across the country using the EDUSAT satellite network. This figure has been drawn with the reference from Mr. Rahul Deshmukh, Web-coordinator, CDEEP. In this figure, the Remote centers are connected by the EDUSAT satellite network. The remote centers are basically Student's Terminal which are used by the students to receive and watch the live transmission of courses. IIT Bombay also has its own Student's Terminal now. The Teacher's terminal and the Video server at IIT Bombay are also linked to the EDUSAT satellite network.

EDUSAT or GSAT3 is a satellite launched by ISRO (Indian space research organization) in the year 2004. This satellite is the first satellite which is exclusively built for educational sector. This is meant to meet the need for interactive satellite based distant education programme such as the CDEEP.

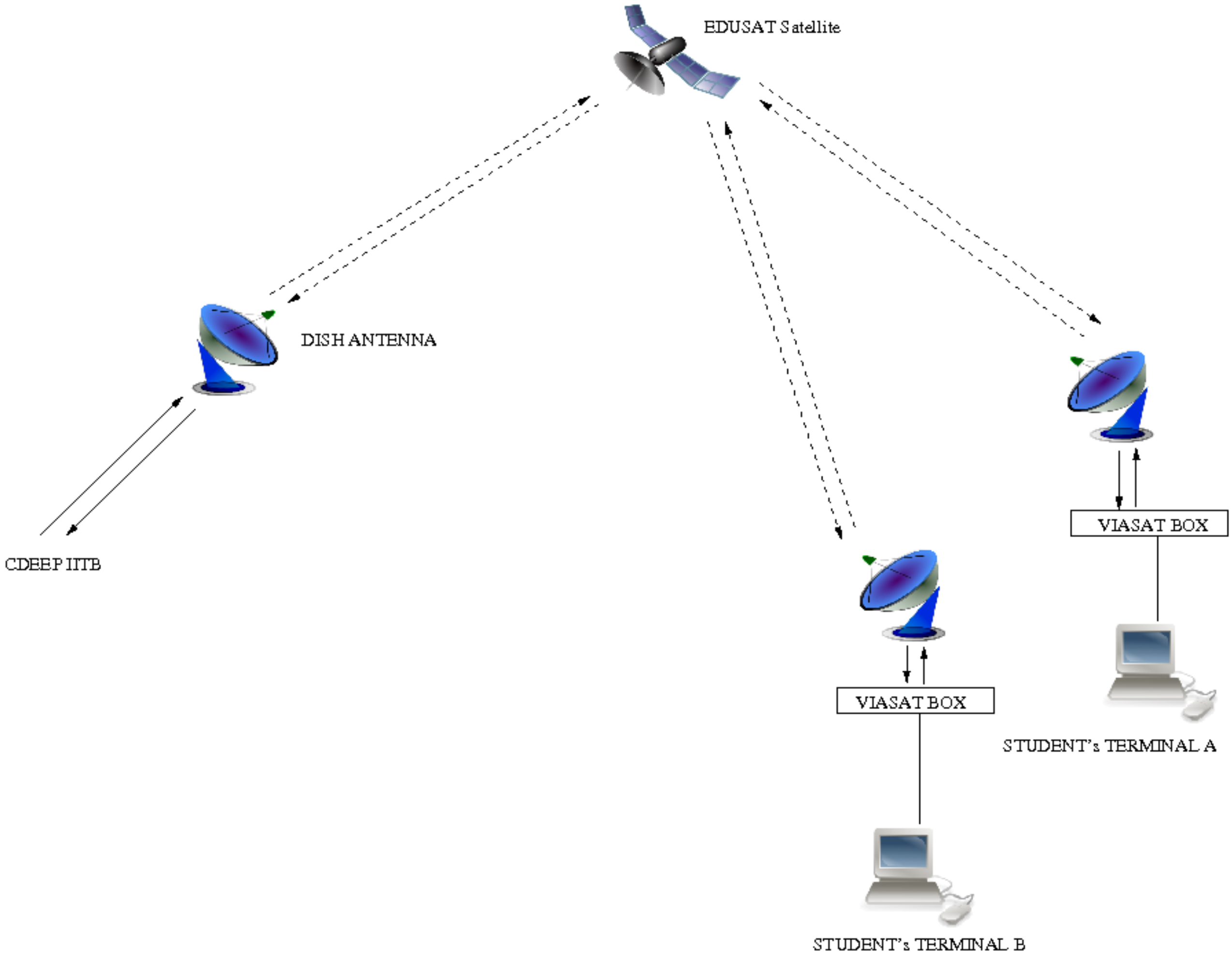


Figure 2.2: CDEEP network structure using EDUSAT Satellite.

The 3 separate type of applications used by the CDEEP EDUSAT satellite network are the Video Server, Teacher's Terminal and the Student's terminal. The Video Server gets the RAW audio/video feed directly from the hardware via standard windows driver. It then encodes and transmits the live audio/video to the multicast server in the EDUSAT satellite network. The application has many settings for the type of video that must be transmitted. The Teacher's Terminal, is for the teacher so that interaction with the remote centers is possible. It can monitor which all remote centers are currently viewing the course. If there is center which wants to ask a question, the notification will be notified by the application. The video transmitted by the students terminal at the remote centers can also be received in this terminal. The Student's Terminal is used at the remote centers. This allows the remote centers to receive the live course transmissions. If students have any questions, the Teachers terminal can be notified by this application. This application has the capability to transmit as well as receive audio/video responses.

Chapter 3

Related Work

3.1 Multicast Networks

IP multicasting is a network technology for efficiently delivering information to a group of recipients simultaneously by sending only one copy from the sender and splitting the information to multiple destinations only when required. In a unicast network a separate copy of information has to be sent to each of the destinations. This makes unicast transmission less efficient while transmitting data that has the same information to multiple destinations as the same data has to be transferred through the network making the network congested. The IP multicast address are in the range 224.0.0.0 to 239.255.255.255. The receivers of a group join the group by a protocol called the IGMP (Internet Group Management Protocol).

3.2 Multicasting on Unicast only Networks

A literature survey on the problems similar to ours was done. A few that were quite relevant to our particular application on CDEEP network is mentioned below.

3.2.1 Example of Mutlicasting on Unicast Network

In a paper on mbone webcast by Milan nikolic , dan hoffman and ljiljana trajkovic [7] they describe about the webcast implementation details and the data they had collected during a multicast session of a conference held in Vancouver in 2001. The Audio and video signals were multicast and used multimedia conferencing tools. They had implemented and setup the multicast session. They conducted the Mbone test session in the Communication Networks Laboratory (CNL) at Simon Fraser University (SFU). They discuss the non-availability multicast routing enabled routers in between the transmission and receiving end points. They further say that one approach to establishing Mbone sessions from such a campus is to make use of tunneling. That is tunneling from the source machine till the point where Internet Mbone is available. This machine will require a DVMRP (Distance Vector Multicast Routing Protocol).

This tunnel should tunnel IGMP inside IP packets that can be sent across a unicast network or multicast disabled networks. In order to achieve this they have used the mrouted program on two workstations running Unix based FreeBSD operating system.

They say in their paper that the idea was to make the local machine which had multicast applications to send their multicast traffic through the DVMRP tunnel to a machine on the other end of the tunnel that was in the remote network and from there the multicast traffic routed locally. This was done so that the routers that were not multicast enabled to not make a problem for the multicast traffic. They transmitted the audio and video signals to two separate multicast servers.As shown in the figure, their setup they had four different networks. One was the IFSA conference site at vancouver, the second the Telus Network to which the IFSA network was connected via an ADSL modem, third the SFU university network and fourth the Bcnet GigaPOP network which was the Mbone network. They made an IP-in-IP tunnel from a FreeBSD machine in conference site to another FreeBSD running machine in the SFU network. A DVMRP tunnel from this FreeBSD machine to a machine in the Bcnet network which they called jade.bc.net. This machine was connected to the Mbone cloud via a gateway.

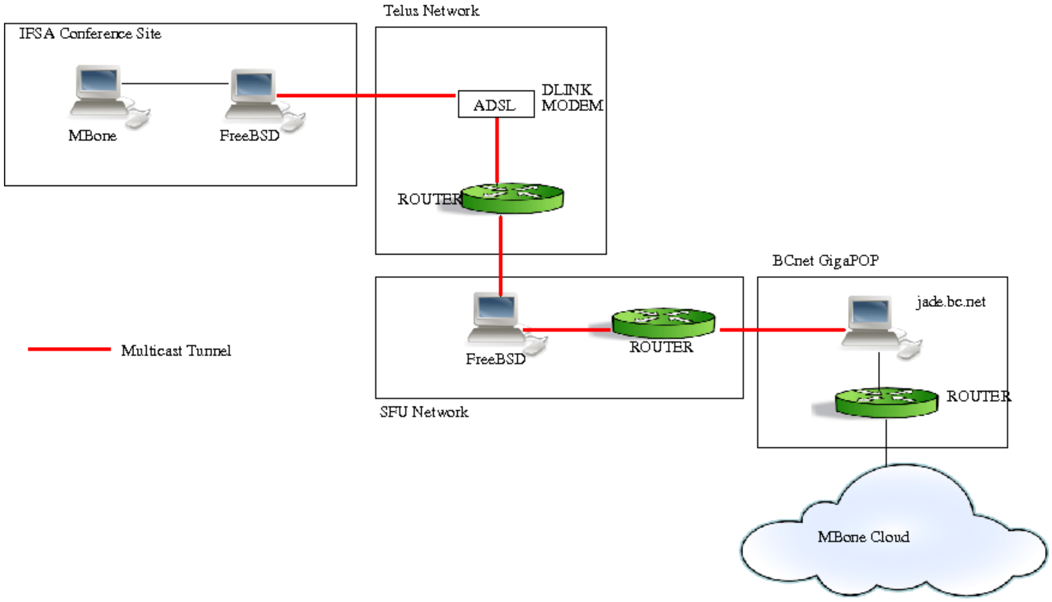


Figure 3.1: Network setup for Mbone webcase at IFSA Conference site [Ref 1].

The setup they have used is very much similar to the one that we have. We have a campus network which is does not support multicast. Also the KReSIT building has the EDUSAT network which is a multicast network. The difference here is that they have the who network setup through the web whereas we have a multicast network though the EDUSAT satellite. But as far is the network at layers other than the physical layers are very much similar.

3.2.2 Improving Effieciency of Hibrid Networks

Lots of related works have been already done to solve the issue of unicast-multicast problem. The concept of unicast-multicast gateway in not new. The reflector [11] is introduced in the RTP and RTSP standards. A few examples of publicly available implementations such applications are mrouted, Multi-session Bridge (MSB), reflector, RTPtrans, UCL Transcoding Gateway (UTG), LiveGate and Reflex.

These applications try to tackle problems mainly of multicast connectivity to unicast participants also. Some applications even do give transcoding features which allows to solve some bandwidth issues. Another instant of already existing reflector is the Darwin Streaming Server which is an open source project from Apple Computers Inc., which includes a multicast-unicast reflector in its RTSP server.

In another paper published by Tarik Cicic, Haakon Bryhni and Steinar Sorlie suggests the architecture of the unicast-multicast reflector. The function of the reflector are (1) forwarding of data traffic between the unicast and multicast network. (2) forwarding of control packets (3) handling of multicast groups (4) session administration and control. They say that the reflector engine should transmit UDP or RTP/RTCP packets from the multicast groups to the unicast recipients. Also a data packet from a registered unicast address is forwarded to the multicast network.

3.2.3 Application Layer Multicasting

Peter Parnes and Kre Synnes and Dick Schefstrm in their paper [9] discuss how a lightweight application level multicast tunneling can be used for sending and receiving multicast data through a underlying unicast network. This applications level multicast must be run on two ends of the tunnel. The multicast data is channeled through the tunnel to be sent to the other side.

Another paper by E. Amir and S. McCanne and H. Zhang [3] talk application level video gateway. Such an application level gateway could be used for our application. Application layer gateway could come in our use by sending audio/video stream from terminal machines to the gateway and then the gateway forwarding the audio/video stream at the application layer to the multicast network.

Chapter 4

Proposal of Solutions

In order for designing and developing solutions for the given problem we need to study the possible ways of approaching the problem to find a good solution. So having a few possible solutions at hand we could then compare each of them to choose the best on suited to us.

4.1 Extending the EDUSAT Network

One thing that we can do is to extend the existing EDUSAT network. As shown in Figure 4.1, we can easily do this by using switches. This is a very obvious and direct solution. We directly put lan cables with switches to the departments to extend the network. The machines on this extended network will belong to the EDUSAT network. This way no other changes in the softwares or IP address allocation have to be changed. This will allow things to work exactly how they are now but with the extended network. This would be mean that we have to add some more elements in the network. In the figure, we extend the network by connecting the cables for outside the KReSIT building to a switch which belongs to the EDUSAT network. As this is fairly obvious and also involves investment of resources, makes this approach less interesting and does not meet our goal of not adding any new infrastructure.

4.2 Extending the Audio/Video Cables

In this approach we could just use long audio/video cables to make reach the RAW audio/video anywhere in the campus. Figure 4.2 shows a brief diagram about this. The range of good quality cables available in the market today be like in the range of 1020 meters. This is quite small for connecting the departments in the campus which are separated by more distance. Moreover, here the issue of transmitting the courses may be solved beautifully. But when the remote centers have to communicate back, then transmitting back the audio/video through such cables are more than just ridiculous. There will have to be operators sitting in the KReSIT building who would

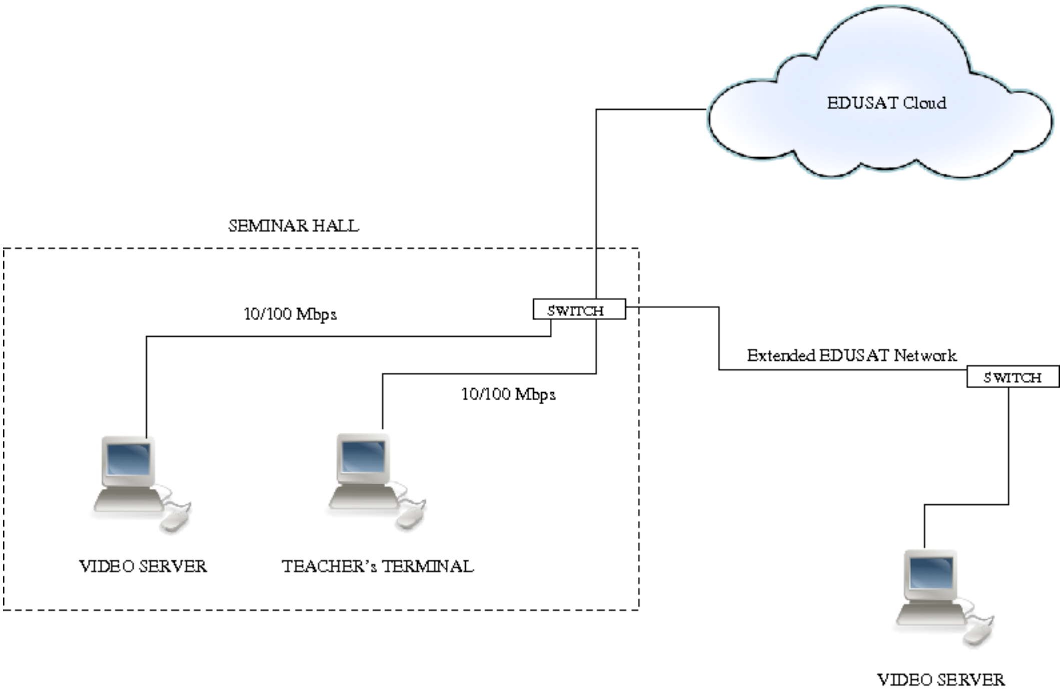


Figure 4.1: Extending EDUSAT network.

be have to communicate by some other means such as telephones to ask whether to relay transmission from which remote centers, etc. They will also have to manually select the audio/video cables corresponding to course that needs to be transmitted onto the satellite network.

4.3 Application Layer Gateway

We can transmit the audio/video as network packets using RTP to forward the transmissions from any department to the KReSIT building. As shown in the Figure 4.3, from from the KReSIT we could receive this video stream and then retransmit this to the EDUSAT network at the application layer. This is basically receiving the packets sent by the video server in the IIT-B network to port and IP as set in the Video server. Then at an application layer we copy the data and retransmit it to the multicast server using the EDUSAT network.

In this approach also since we are developing applications from scratch in order to transmit we will have to do the same for the reception also. This means that there must be operators who will have to relay the audio and video and the departments will have to be always in touch with the operators who are in a different building which could make it highly unreliable solution.

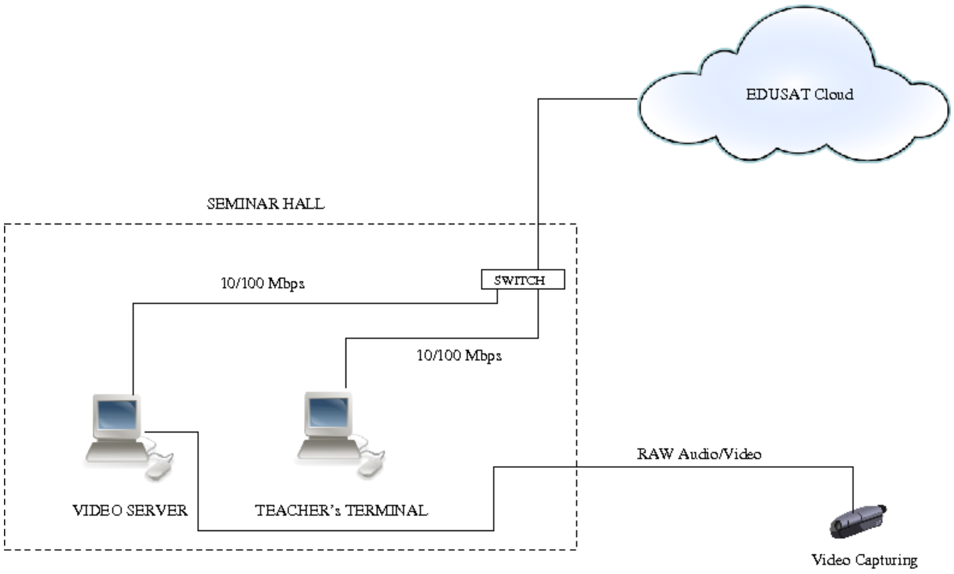


Figure 4.2: Extending Audio/Video cable.

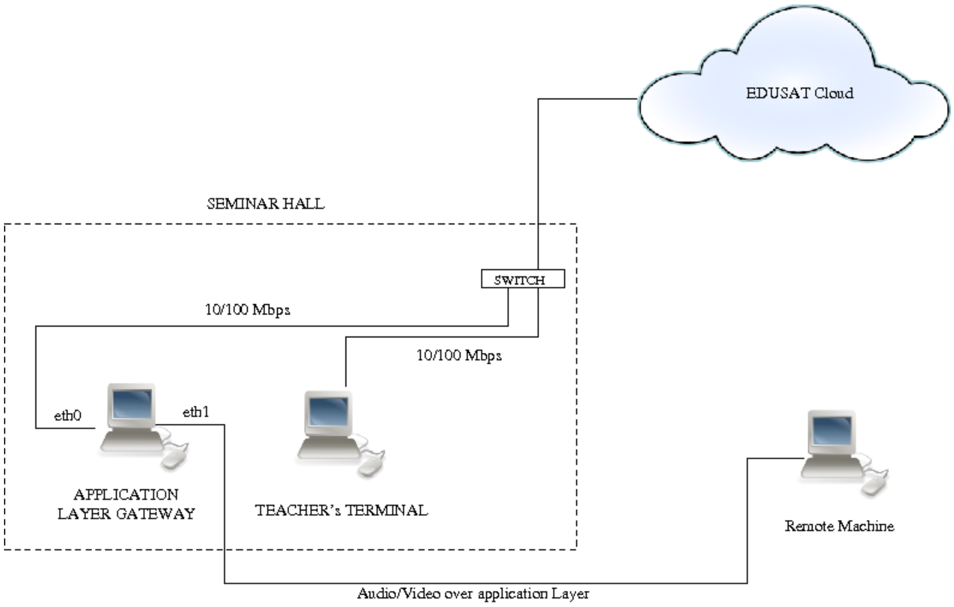


Figure 4.3: Application Layer gateway.

4.4 Transmitting RAW Audio/Video over Network

If we could find a way to transmit RAW audio/video over the network, then a small trick could be done from transmitting audio/video through the EDUSAT network. The proposal is of somehow using the hardware device of a remote machine through the network. The basic idea that motivated me to do this was that hardware resource of a machine could be directly shared over the network. For example , a mass storage device. This means that the hardware of some other machine would be seen as the hardware of its own. This way we could use the capture device of the machine in some department and use that to feed RA audio/video to the Video server software. Figure 4.4 shows how this would look like.

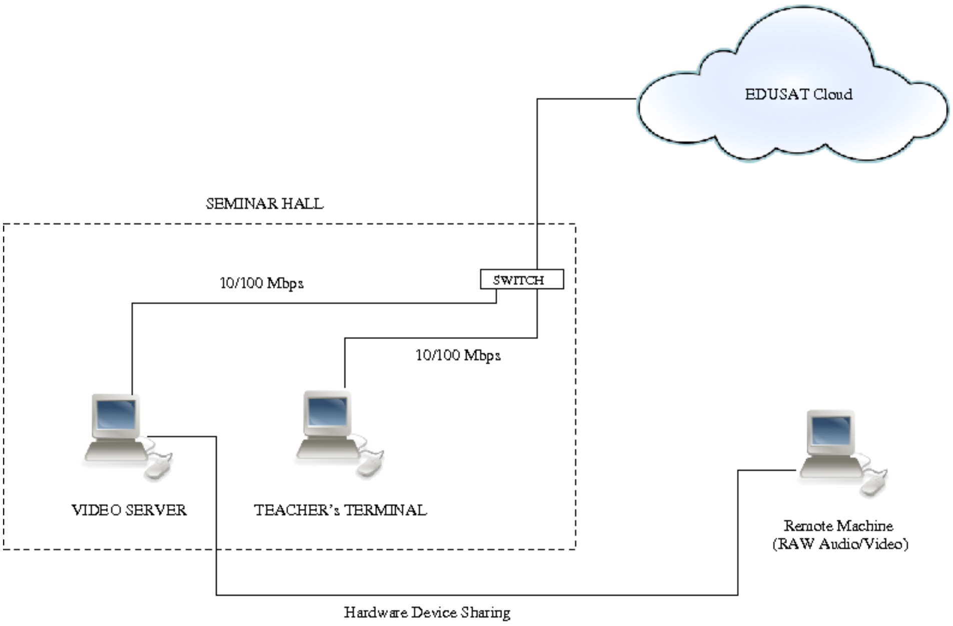


Figure 4.4: Transmitting RAW audio/video over the network.

Like the above solution, even if it really worked, it would solve only the transmission of video lectures. The reception cannot be done this way. But even then we could set up an intermediate machine that would receive the transmission from the remote centers and play them as RAW audio and video. This RAW audio and video will have to be again fed into some other machine whose capture device could be shared to some remote machine.

4.5 Network Layer Gateway with NAT (Network Address Translation)

As above approaches have limitations of not being to able to solve the problem of bi-directional communication, ie we can only solve the problem of transmitting the course to remote centers and not of receiving back from the remote centers, we need a way to find a different approach so that we use the same network at the physical layer and the same applications at the application layer. We have to into some other layers especially the network layer for a better solution so that for the end users and applications, everything is transparent. Even though there are are translations of the packets happening at the network layer, the users must be to feel any difference whether they are in the EDUSAT network or the IIT-B network.

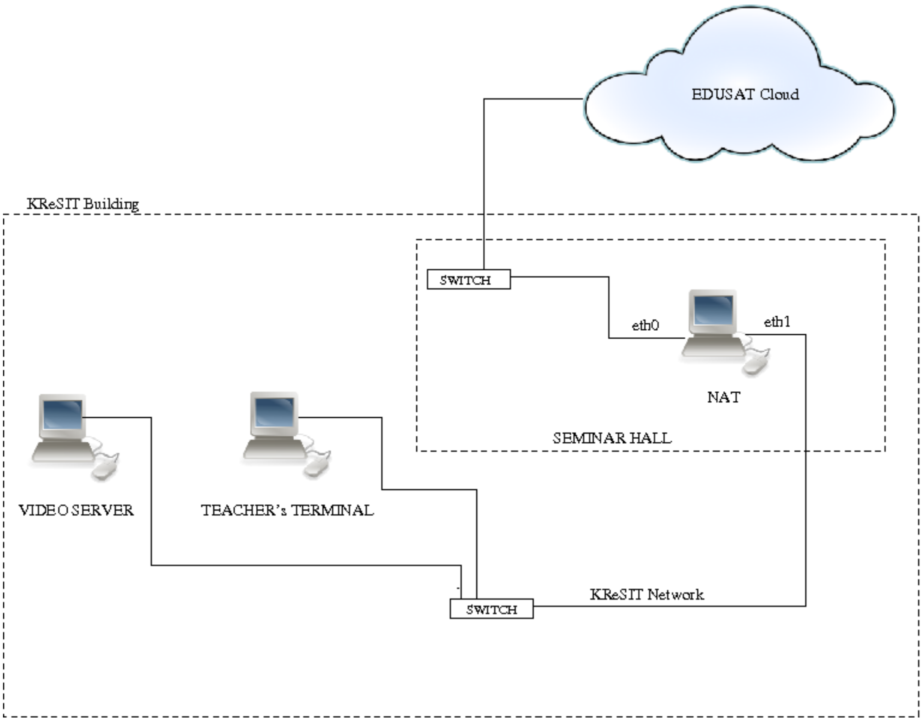


Figure 4.5: Gateway at network layer doing NAT.

Referring to the Figure 4.5, the basic idea is to setup a network layer gateway that would do the job of forwarding and interlinking network packets back and forth between the two unicast and multicast networks.

NAT or Network Address translation is the process of modifying the network address details inside the packet headers while in transit across the network so that a particular address space can be mapped into some other address space. NAT can be used along with IP masquerading to hide an entire address space so that only a single public address is visible to the other users outside the private network.

We setup a NAT (Network Address Translation) machine that would do the work of changing the source and destination address before forwarding the packets to the other network. The IP address of the NAT machine in my setup was 10.254.75.2 (eth0) (refer Figure 4.5) on the private network where Teacher's terminal and Video server are connected and 192.168.0.1 (eth1) which was connected to the EDUSAT network. Then on this machine we setup the NAT, using iptables, etc, such that any packet that originates from eth1 (refer Figure 4.5) and arrives at the NAT machine, forward them by translating the source address to the address of eth0 (10.254.75.2). This means that the packets that go to the multicast server will be from the NAT machine and as if it originated from the NAT machine itself. The advantage is that now the terminal machines can be on another network and still send and receive information from the EDUSAT network. The NAT machine will do the job of making the packets compatible with EDUSAT network. Hence if this is possible then we can place the Video Server as well as the Teacher's terminal anywhere on the network.

4.6 Network Layer Gateway with NAT and Tunneling

The problem in the approach above is that there cannot be a unicast router in between the terminal machines and the NAT machine. We know that if we were to send and receive from anywhere inside the campus using the IIT-B network, the routers of its network which are also unicast come into picture. Refer to Figure 4.6 for this. In the previous approach of NAT the teacher's terminal and the video server were connected to the NAT machine and was in the same network. So they saw the NAT machine as their default gateway. But for a network outside the KReSIT building, the default gateway has to be the gateway each department or building has. So once with such a setting if we send a packet with destination as the multicast server in the EDUSAT network, the routers in the campus do not have information for forwarding those packets. But our goal is to able able to transmit and receive from anywhere in the department. In that case the unicast routers would definitely come in between them.

Hence we will not be able to use approach 3.5 for using outside the KReSIT building.So we need an idea of a packet inside a packet or the concept of tunneling. The idea is to make a tunnel through the IIT-B network cloud. In the Figure 4.6, this way the packets that originate from the Video server are sent to the tunneling machine. The tunneling machine encapsulates the multicast packets and send it to the other end of the tunnel which is the NAT machine. The NAT machine then does a usual NAT and forwards it to the EDUSAT network. The packets coming from the EDUSAT network are also first forwarded into the tunnel by the NAT machine. Then the end terminal machine receives the packets through the tunnel.

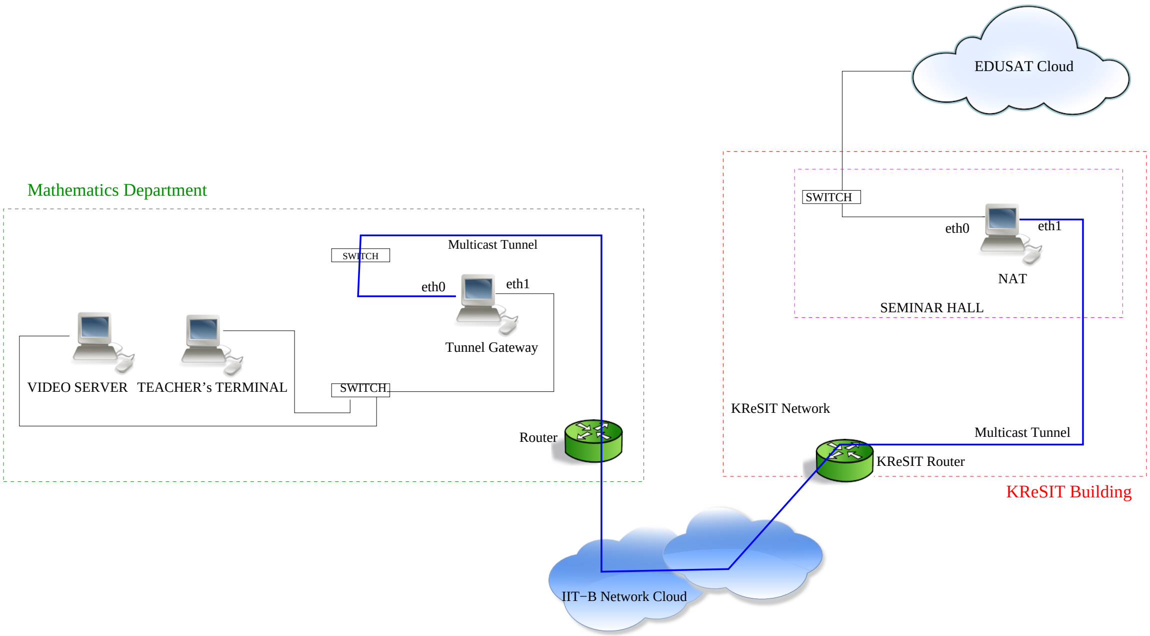


Figure 4.6: NAT with tunneling.

4.7 Network Layer Gateway with Tunneling

We do not specifically need a NAT as mentioned in previous approach while using a tunnel. In the previous approach we had assumed we need only one IP-address in the satellite network to be tunneled. But instead we could use any IP-address (belonging to the EDUSAT network) in the extended network. Since the packets are already being tunneled through the gateways, we could give IP addresses that belong to the EDUSAT network itself to the terminal machines as shown in Figure 4.7. These packets could be forwarded to the EDUSAT network without making any changes.

4.8 Tunneling at Routers

Yet another way to do tunneling is to do this at the routers itself. These routers must support tunneling as well as multicasting. This is very similar to the previous approach. But this could give a performance boost while encapsulating and decapsulating because of doing at routers that have hardware designed specifically to do this job.

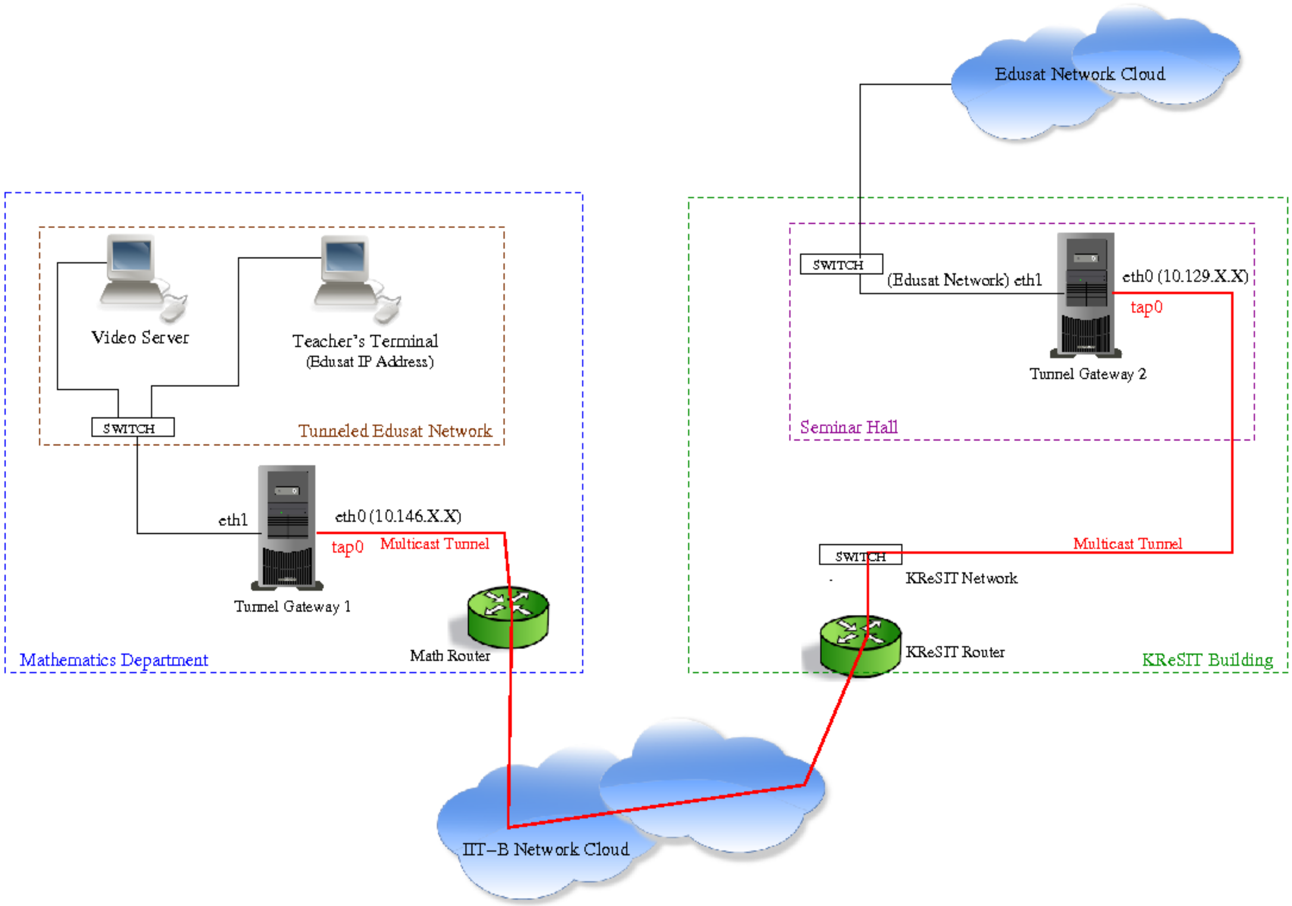


Figure 4.7: Setup for CDEEP using tunnel and bridge.

Chapter 5

Implementation

5.1 Application Layer Gateway

In this implementation there is an application layer gateway as in Figure 4.3. Consider a machine in some department outside the KReSIT (say hostel 12).Lets call this machine Modified Video Server with IP address 10.12.11.3. Another machine is setup in seminar hall of KReSIT building which is an application layer gateway. This machine has two network interfaces. One belonged to IIT-B network with IP address 10.129.154.150 and the other belonged to the EDUSAT network with IP address 10.254.75.2 (IP address of the Video Server).

I implemented the applications in JMF (Java Media Framework). JMF is library for java applications and applets to enable audio and video. Using this package, multiple video formats can be captured, played, streamed or transcoded. Here we use this because we want the audio and video to be transmitted to the application layer gateway. JMF has library in order to do these functionalities. At the gateway, we have to receive, transcode and stream it to the multicast server. The Modified Video Server had a server developed in Java that would stream audio and video to the machine (Application layer gateway). Here this machine had a client and a server both running at the same time. The client receives the audio and video from the Modified Video Server. It copies the data and then re-encode it again to send it to the mulicast server on the other interface.

5.2 Transmitting RAW Audio/Video over Network

For this readily available software packages that could possibly be used. Many of these was made for sharing USB driver over the network. I could use this to share the Video card of the Video Server machine and then use this on the EDUSAT network machines to receive live streaming and then do usual transmission.

One of the softwares that I used was FabulaTech USB over Network Version

4.2. It has two separate applications. One , called the Device Workstation, has to be installed on the server side , ie where the capture device is actually there. The other , called the Remote Workstation, has to be installed on the machine which will be using the remote capture device as if it were its own.

So once this is done we can use the capture device of the remote machine. And whatever is fed into the device of the remote machine must come into the applications using the using the shared capture device.

5.3 Network Layer Gateway with NAT

NAT or Network Address Translation can be done using iptables in linux quite easily. The machine simply does a network translation of the packets it receives. In our setup we made a separate network with IP addresses in the range 192.168.0.0/24. The Video Server and the Teacher's terminal are connected on this network. The NAT machine has 2 network interfaces. One belongs to the this private network. In our experiment this interface card (eth1) was 192.168.0.1. The other interface card (eth0) had IP address 10.254.75.2 which belonged to the EDUSAT network. The iptables were so configured so that any packet coming from the eth1 interface was done with a source address NAT to 10.254.75.2 before sending to the EDUSAT network. The packets that were coming in the other direction, ie, arriving on the eth0 interface from the EDUSAT network had a destination NAT to 192.168.0.0/24 correspondingly based on the iptables. The following is the key command for doing a NAT, referring to the setup as in Figure 4.5

```
sudo iptables -A FORWARD -i eth0 -o eth1 -s 192.168.0.1/24
```

Hence we moved the Video Server and the Teacher's terminal out of the EDUSAT network into another private network. So they where no more directly on the EDUSAT network. The is supposed to NAT packets from this private network so they appeared to the multicast server as they had come from its own EDUSAT network.

5.4 Network Layer Gateway with NAT and Tunneling

In this implementation we have to use the NAT as in previous proposal. This time one interface of the NAT (eth1) will have IP address that belongs to the KReSIT network say 10.129.154.203. The other (eth0) as in previous 10.254.72.2 belongs to the EDUSAT network.

In the Figure 4.6, the NAT machine has two network interfaces eth0 and eth1. Eth0 is connected directly to the EDUSAT network and has IP address 10.254.75.2.

The other interface has IP address that belongs to the KReSIT building, 10.129.154.205. In the figure, the mathematics department is outside KReSIT. The tunnel gateway is the gateway for the machines in the department which wants to communicate with the EDUSAT network. The tunnel gateway also has two interfaces. He eth0 belongs to the IIT-B network with IP address 10.105.11.200 and the eth1 belongs to the private network with IP address 192.168.0.1. Now the packets that come from the teacher's terminal or the Video server goes to the tunnel gateway where it is encapsulated with IP-in-IP and the sent to the NAT machine. Here the packets are decapsulated and then translated with the source IP 10.254.75.2 and transmitted. In reverse the packet from EDUSAT network arrives at the NAT machine which are translated with source IP 10.129.154.205 and encapsulated and sent to the tunnel gateway. The tunnel gateway decapsulates them and forwards it to the teacher's terminal and video server.

5.5 Network Layer Gateway with Tunneling

There are many ways in which a tunneling can be made.

GRE (Generic Routing Encapsulation) tunnel GRE is a tunneling protocol [10] developed by Cisco system that can be used for tunneling various network protocols inside of IP tunnels creating a virtual point-to-point link between routers or computers.

mTunnel

The mTunnel [8] is an application that tunnels multicast packets over an unicast UDP channel. Several multicast streams can be sent over the same tunnel while the tunnel will still only use one port.

OpenVPN

OpenVPN [2] is a free and open source software application for creating secure point-to-point routed or bridged configurations for making a virtual private network (VPN) through an existing network connection. It uses SSL/TLS security for encryption.It was originally written by James Yonan and is published under the GNU General Public License (GPL) [13].We will use OpenVPN to setup the tunnel.

5.5.1 Multicast Routing Daemon

For the routing the multicast packets through the tunnel, one of the options is to use a multicast routing daemon. One such daemon is the pimd or protocol independent multicasting daemon. This application runs a multicast daemon over the virtual tunnel interface and the EDUSAT interface. Hence the multicast stream could be routed according to the Internet Group Messaging protocol for multicasting or IGMP.

mrouted [1] is another is a routing daemon that can forward IP-multicast packets through routers that support IP multicast. The routing protocol implemented by mrouted is the Distance-Vector Multicast Routing Protocol (DVMRP) [12].

mrouted is available for linux versions. But mrouted is not a free software. Because of this it is not distributed via normal linux distributions. It has to be downloaded and installed.

5.5.2 Bridging

Instead of routing from the virtual tunnel interface to the physical EDUSAT interface, we could just use a small technique of bridging [5] the virtual interface and EDUSAT interface at the Layer 2 or MAC layer. This has an advantage for this specific application as we want the packets to be forwarded across the gateways as soon as possible rather than wait for an application to decide where to forward according to IGMP. As the bridging is so reliable and very easy to setup, bridging would be a better option.

Make an Ethernet-tunnel between EDUSAT Satellite network and IIT-B campus network. This way we tunnel the Ethernet-packets through the IIT-B campus network cloud. Basically the packets must get exchanged between the end terminals (the end terminal applications used for conducting live sessions) without the end terminals knowing about this additional travel through the IIT-B campus cloud. This tunnel will allow us to transmit and receive IP-multicast packets through the unicast only IIT-B campus network.

Setup and Configuration

The Figure 5.1 shows an overall view of the setup we are going to make. Here we tunnel from the Math Department to the KReSIT building at IIT Bombay. The two big dotted boxes shows each of these departments. Each one has a department router or gateway, green pie in the figure, which is usually addressed as X.X.1.250. The cloud on the top represents the EDUSAT Satellite network that was originally isolated from IIT-B network. The two machines Tunnel Gateway 1 and Tunnel Gateway 2 are the machines that we will be setting the tunnel on. Each of these machines have two network interfaces each marked as eth0 and eth1. One of these gateway will be at the Seminar Hall in KReSIT and the other in the Math department. The cloud at the bottom shows the campus network, the details of which we are not concerned within our setup. We will be tunneling through this cloud. The tap0 is the virtual interface or tunneling interface that we will make during the setup. The end terminals, Teacher's terminal and video server, etc. will then be connected to the eth1 interface of the Tunnel Gateway 1 in Math department via a switch if needed. This way we make the EDUSAT network extended to the IIT-B campus network without any additional cabling. You can refer [4] for details.

Requirements

Two PC's (the Tunnel Gateways in Figure 5.1) with two network interfaces (NIC's) each. We have done the setup on Ubuntu 9.04 (Jaunty Jackalope). We will explain the setup for this operating system. If you are using any other operating system then you might have to do corresponding changes specific to it. Anyway the concept is just the same. The same setup might work on other versions of Ubuntu and Debian operating systems as well. We would recommend Ubuntu 9.04, as it is very stable.

Physical Setup

Let us name the two PC's as PC-Server and PC-Client (Tunnel Gateway 1 and 2 in Figure). This is because openvpn configures a server and clients connect to the server.

PC-Server

PC-Server should have two interfaces eth0 and eth1 (do an ifconfig to verify) as shown in Figure 5.1. Let PC-Server be connected to EDUSAT Satellite network (so must be in the KReSIT building) through the eth1 interface and connected to the KReSIT network through eth0 interface.

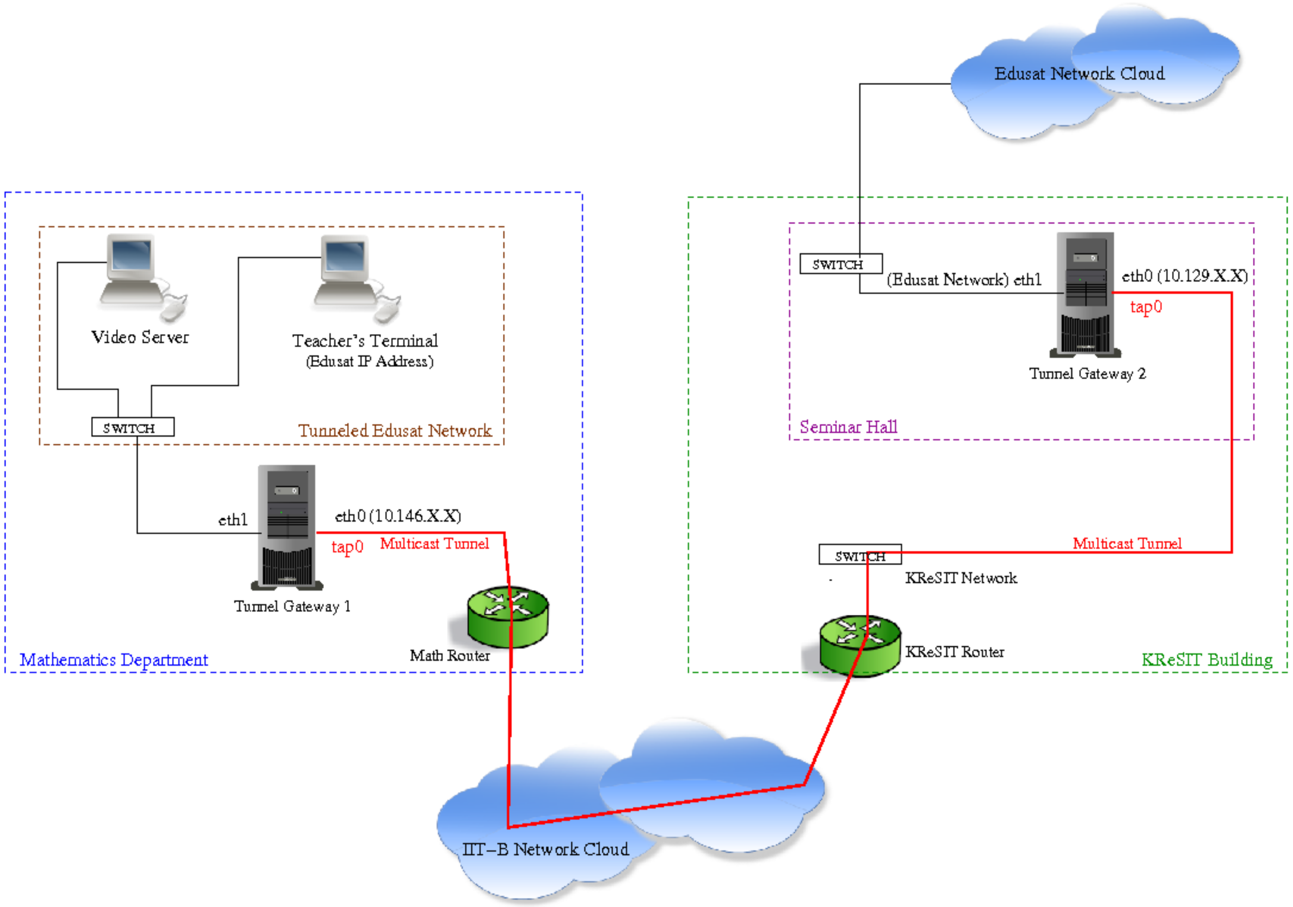


Figure 5.1: Setup for CDEEP using tunnel and bridge.

Give a static IP-address to eth0 (KReSIT IP address allocation scheme if in KReSIT). Let this be $IP-Addr-Server$. You do not need to give any IP-address to the eth1 interface.

eth0 : KReSIT network (IP Address: $IP-Addr-Server$)

eth1 : EDUSAT Satellite network (no need of IP Address)

PC-Client

Similarly let PC-Client (which may be anywhere inside the IIT-B campus) be connected to IIT-B network through the eth0 interface. You can connect the end terminals that run applications for conducting live sessions (Teacher's terminal or Video server) to the eth1 interface (via a switch if you want to connect more than one machine).

Give a static IP-address to eth0 (depends on where the machine is kept inside campus). Let this be $IP-Addr-Client$. Again you do not need to give any IP-address to the eth1 interface.

eth0 : IIT-B Campus network (IP Address: $IP-Addr-Client$)

eth1 : Connected to end terminals (no need of IP Address)

Installing required packages

Install openvpn, bridge-utils and ssh on both PC-Server and PC-Client.

```
sudo apt-get install openvpn bridge-utils ssh
```

Now Ethernet-tunnel between PC-Server (openvpn server) and PC-Client (openvpn client) has to be setup. For this the keys for setting up the tunnel have to be generated. They are usually found (in Ubuntu 9.04) in

```
/usr/share/doc/openvpn/examples/easy-rsa/2.0/
```

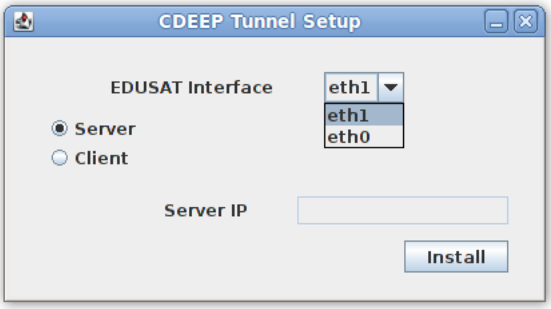
Connecting the Terminal Application Machines

Once the above settings are made, what remains is only to connect the machines that are used for transmitting and receiving audio/video stream for conducting video lectures to the gateway machine. There is no need to change any configuration in these machines. If you intend to connect more than one such machines, it can be done by connecting them via a standard switch to the gateway's tunneled EDUSAT interface.

5.5.3 Java Application for Automating Tunnel Installation

An application was developed in Java for automating the setting up of tunnel customized for the needs of CDEEP. As shown in Figure 5.2 the user has to choose whether he wants the machine to be acting as a server for the gateway or client. He also has to choose which interface he would like bridge. This interface is the interface that would be connected to the EDUSAT satellite both on the client as well as the server side.

The application was developed using Java Swing for the graphical user interface.



CDEEP Tunnel Setup

EDUSAT Interface: eth1

● Server

○ Client

Server IP:

Figure 5.2: Java application for automating setup.

Chapter 6

Evaluation and Performance Measurement

6.1 Testing the Application Layer Gateway

The application layer gateway that was developed in Java was deployed for testing. The audio and video was being received at the gateway machine. This was then also being re-encoded and sent to the multicast server. But the multicast server did not receive the audio or video.

May be the encoding, frame rate, protocol used, and parameters such as that where not correct.

6.2 Testing Sharing of Hardware over the Network

For transmit RAW audio/video over the network softwares that allow sharing of hardware devices over network was tried. Most of the softwares that I got where for sharing USB devices and especially mass storage devices. Even though these softwares seemed to work for mass storage devices, it was not so friendly with video capture devices such as the webcam.

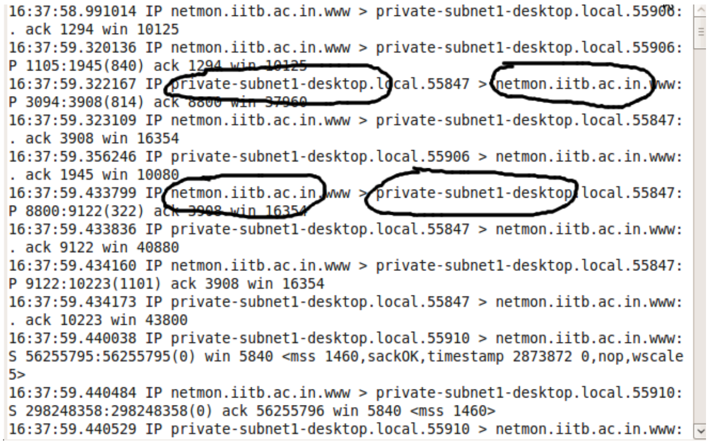
It was possible to share the hardware of a webcam. The hardware appeared on the remote machine as if it were its own. I even installed the device drivers for it. But when it came to the actual capturing or controlling of the device, the device did not respond at all.

May be we need to build a custom driver if we where to make this happen.

6.3 NAT on EDUSAT Satellite Network

The results where really optimistic here. Even though the Video Server and teachers terminal where now on some other separate private network, NAT machine was able to reach the packets to the multicast server successfully.

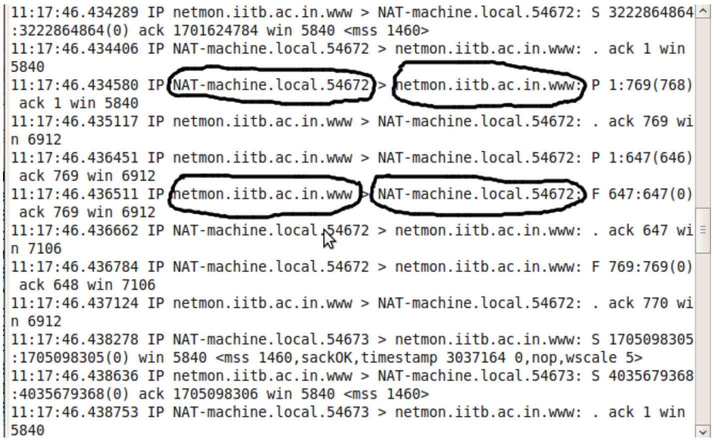
This also had the advantage that nothing of the Teacher's Terminal or Video Server application configurations had to be changed. It was completely transparent to these applications. Improvement in this method would be the most suitable solution to the given problem. In the Figure 6.1, we can see that the machine connected to the NAT sends packets in the normal way. In this case from itself to the public machine netmon.iitb.ac.in.



```
16:37:58.991014 IP netmon.iitb.ac.in.www > private-subnet1-desktop.local.55906:
. ack 1294 win 10125
16:37:59.320136 IP netmon.iitb.ac.in.www > private-subnet1-desktop.local.55906:
P 1105:1945(840) ack 1294 win 10125
16:37:59.322167 IP private-subnet1-desktop.local.55847 > netmon.iitb.ac.in.www:
P 3094:3908(814) ack 8800 win 37960
16:37:59.323109 IP netmon.iitb.ac.in.www > private-subnet1-desktop.local.55847:
. ack 3908 win 16354
16:37:59.356246 IP private-subnet1-desktop.local.55906 > netmon.iitb.ac.in.www:
. ack 1945 win 10080
16:37:59.433799 IP netmon.iitb.ac.in.www > private-subnet1-desktop.local.55847:
P 8800:9122(322) ack 3908 win 16354
16:37:59.433836 IP private-subnet1-desktop.local.55847 > netmon.iitb.ac.in.www:
. ack 9122 win 40880
16:37:59.434160 IP netmon.iitb.ac.in.www > private-subnet1-desktop.local.55847:
P 9122:10223(1101) ack 3908 win 16354
16:37:59.434173 IP private-subnet1-desktop.local.55847 > netmon.iitb.ac.in.www:
. ack 10223 win 43800
16:37:59.440038 IP private-subnet1-desktop.local.55910 > netmon.iitb.ac.in.www:
S 56255795:56255795(0) win 5840 <mss 1460,sackOK,timestamp 2873872 0,nop,wscale
5>
16:37:59.440484 IP netmon.iitb.ac.in.www > private-subnet1-desktop.local.55910:
S 298248358:298248358(0) ack 56255796 win 5840 <mss 1460>
16:37:59.440529 IP private-subnet1-desktop.local.55910 > netmon.iitb.ac.in.www:
```

Figure 6.1: Tcpdump at terminal machine.

Also it receives packets from this public machine even though it is not directly connected to the public network. In Figure 6.2, at the NAT machine we can see that all packets go out from it as though it itself is sending them, in this case as NAT-machine.local. The received packets are address translated because of which we saw such an output in Figure 6.1.



```
11:17:46.434289 IP netmon.iitb.ac.in.www > NAT-machine.local.54672: S 3222864864
:3222864864(0) ack 1701624784 win 5840 <mss 1460>
11:17:46.434406 IP NAT-machine.local.54672 > netmon.iitb.ac.in.www: . ack 1 win
5840
11:17:46.434580 IP NAT-machine.local.54672 > netmon.iitb.ac.in.www: P 1:769(768)
 ack 1 win 5840
11:17:46.435117 IP netmon.iitb.ac.in.www > NAT-machine.local.54672: . ack 769 wi
n 6912
11:17:46.436451 IP netmon.iitb.ac.in.www > NAT-machine.local.54672: P 1:647(646)
 ack 769 win 6912
11:17:46.436511 IP netmon.iitb.ac.in.www > NAT-machine.local.54672: F 647:647(0)
 ack 769 win 6912
11:17:46.436662 IP NAT-machine.local.54672 > netmon.iitb.ac.in.www: . ack 647 wi
n 7106
11:17:46.436784 IP NAT-machine.local.54672 > netmon.iitb.ac.in.www: F 769:769(0)
 ack 648 win 7106
11:17:46.437124 IP netmon.iitb.ac.in.www > NAT-machine.local.54672: . ack 770 wi
n 6912
11:17:46.438278 IP NAT-machine.local.54673 > netmon.iitb.ac.in.www: S 1705098305
:1705098305(0) win 5840 <mss 1460,sackOK,timestamp 3037164 0,nop,wscale 5>
11:17:46.438636 IP netmon.iitb.ac.in.www > NAT-machine.local.54673: S 4035679368
:4035679368(0) ack 1705098306 win 5840 <mss 1460>
11:17:46.438753 IP NAT-machine.local.54673 > netmon.iitb.ac.in.www: . ack 1 win
5840
```

Figure 6.2: Tcpdump at NAT machine.

6.4 Tunnel with Multicast Routing Daemon

The first attempt to make the daemon run was made on a machine running ubuntu and the daemon was mrouted [1]. As explained above the daemon implements DVMRP. But since it is not a free software, it does not come along with linux distributions. It has to be downloaded and installed separately. But the kernel showed some bugs and failed to install.

mrouted initially came on FreeBSD operating systems. In the hope that it could be easily installed on that operating system, an attempt was made. But the kernel had to be recompiled with the support for multicast packet forwarding by changing a value in the kernel configuration file

```
options MROUTING
```

The next routing daemon used was pimd which implements Protocol Independent Multicast routing. This is available in linux distributions and is also free software. The tunnel was setup and this daemon run on the tunnel interfaces. This did not work for all streams. Using a multicasting daemon to run over the virtual interface did not work for all types of multimedia streams. If the sending stream where RTP with audio and video separate on different ports, it worked. But if the stream was send as a UDP stream , it did not work. Also it took a long time for the stream to actually go through the tunnels as it takes time for the daemon to add the multicast group information to its table and then forward it upon receiving a joining request.

But a more reliable way would be to just bridge the tunnel interface with the physical interface so that it becomes very easy as well as the traffic between the tunnels is forwarded almost instantaneously. This is possible because the tunnel we use is an ethernet tunnel. Hence they can be bridged at layer 2 as any usual physical interface. Advantage of this is that no routing table has to be setup or no need of making any changes on the end terminal machines that will be running applications. So this method would also make it completely transparent to the end machines. Next section we describe how the bridge performs on the tunnel.

6.5 Performance Evaluation of Tunnel with Bridge

Tunneling and then bridging the virtual interface with the EDUSAT physical link is a very convenient and easy way of extending the EDUSAT satellite network through the campus network. Since this works, this was put on an experiment for testing for its performance through the campus network. In a paper by Inoue, Takeru and

Kurebayashi, Ryosuke [6] they did their formulation of tunneling impact on multicast efficiency. Here we use our own ways to measure the efficiency of the tunnel since our application is a very specific one. In next subsection, we describe how we put the tunnel alone to a efficiency test and measure its performance. For this a setup with four machines isolated from other networks was done.

6.5.1 Controlled Experiment

Figure 6.3 shows how the setup was made so as to measure the performance of the tunnel.

Six different configurations were made by using combinations of no-tunnel, bridged-tunnel, routed-tunnel, cipher-tunnel. In the Figure 6.3 there are four machines A, B, C and D physically connected in the same way for all configurations. Machines A and D have gateways B and C respectively. Machines B and C each had two physical LAN cards. In all the configurations the performance of the tunnel was measured by sending audio/video stream from machine A to machine D. As shown in Figure 6.3, in configuration (a) Only simple routing was done at Layer 3 using routing table and no tunneling was used. In (b) Instead of routing, the two interfaces at both the gateways were bridged at Layer 2. In (c) a tunnel was setup between machines B and C and routing table was set to route through this tunnel. (d) is similar to (c) except that the tunnel interface/virtual interface was bridged with the interfaces connected to machines A and D. Configurations (e) and (f) are same as (c) and (d) except that (e) and (f) had an encrypted tunnel setup.

The following table 6.1 shows a few performance parameters of using a bridge over routing. It shows the parameters of the received audio/video stream at the receiver side for all the configurations mentioned above. The ping delay was also measured.

The observation that was made here was that all the configuration had almost negligible differences. But speaking more precisely, for all the configurations the parameters calculated out of the audio/video stream was more or less same. But the differences could only be seen in the ping delay for various configurations. The ping delay was more by a small fraction, as shown in table 6.1, when using a tunnel. Also ping delay still increased by small fraction if the tunnel were to be encrypted/cipher tunnel. Comparing bridging and routing, bridging had a bigger ping delay than routing. Anyway, since the values are in the range of one-hundredth of a millisecond, these are just negligible for our specific application as the delay through the satellite network would anyway be much more.

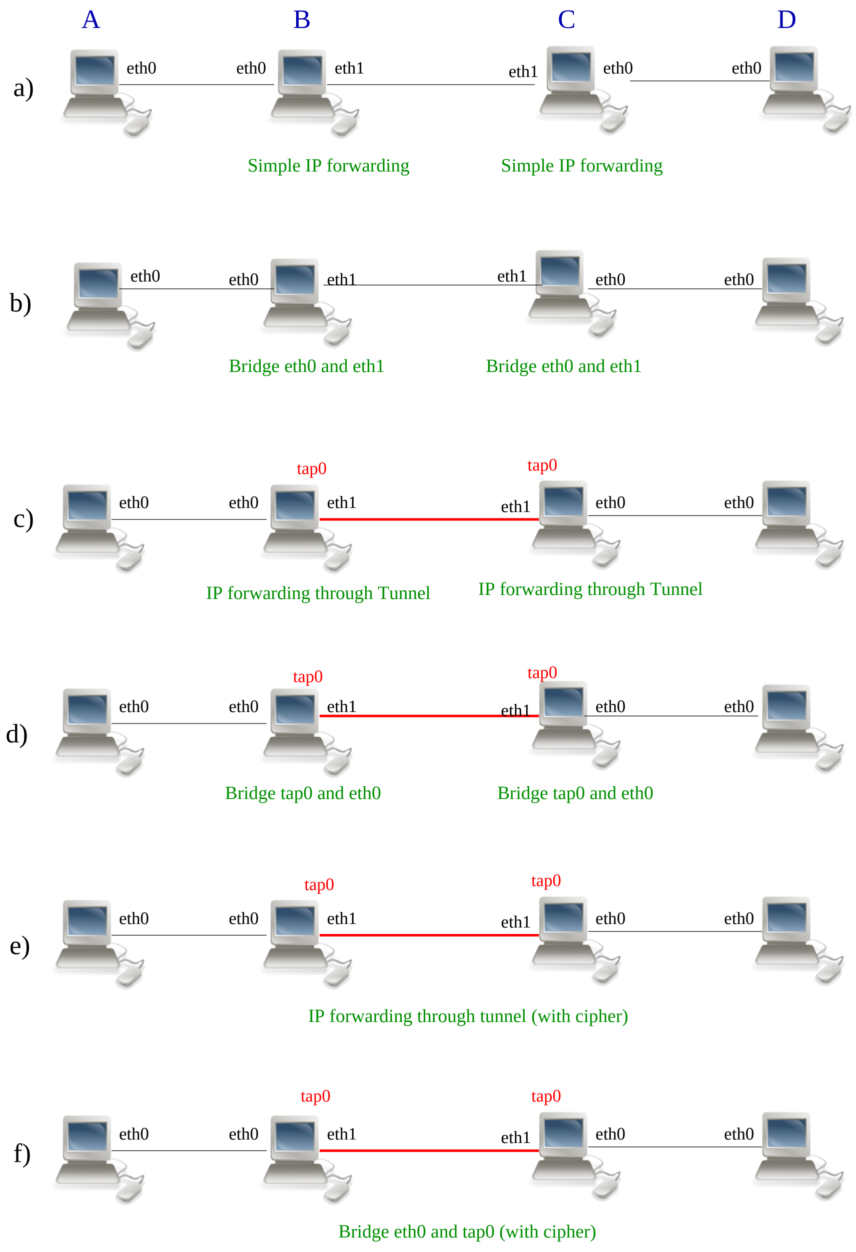


Figure 6.3: Experimental setup for measuring performance of the tunnel.

6.5.2 Experiment on EDUSAT Satellite Network with Tunnel

To test the impact of the tunnel through the campus network, a live experiment was conducted while a course transmission was happening through the EDUSAT satellite network. This was done by making an ethernet tunnel using OpenVPN from the 4th Floor CDEEP Lab in KReSIT to the Hostel-12 in IIT-B through the campus network. For this a machine with two LAN cards was setup in the 4th Floor KReSIT and a

Stream Property	Without tunnel		With tunnel			
			No cipher		Cipher	
	No bridge (ms)	Bridge (ms)	No bridge (ms)	Bridge (ms)	No bridge (ms)	Bridge (ms)
Ping delay	1.062	1.065	1.113	1.117	1.121	1.125
Jitter	2.4	2.4	2.41	2.38	2.41	2.4
Minimum	0.007	0.007	0.007	0.007	0.007	0.007
Maximum	15.65	15.7	15.7	15.6	15.73	15.73
Average	7.01	7.01	7.0	7.0	7.01	7.0
Packet Loss	0.4	0.4	0.45	0.5	0.4	0.4

Table 6.1: Tunnel Performance with various configurations

similar machine in Hostel-12. A tunnel was made between these two machines. The experiment for measuring the quality of audio/video stream through the tunnel was done for the working hours of complete day when the transmission was going on live from 3rd Floor KReSIT seminar hall which has a direct connection to the EDUSAT satellite network.

This experiment was done to find out if tunneling through campus has any effect on the quality of audio/video stream due the network traffic in the campus. The graph 6.4 shows the variation in the parameters, jitter, minimum jitter, maximum jitter, average interpacket delay and ping delay, that were measured for a day.

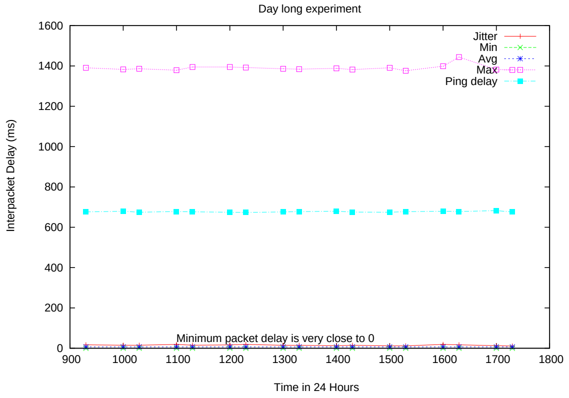


Figure 6.4: Interpacket delay for a day.

As seen in the graph 6.4 there is almost no significant variations in any of the parameters that were measured. The variation in jitter of the audio/video stream is approximately from 11ms to 19 ms. This variation in jitter is for a whole day. Also this variation in jitter is not only due to the variation in the network traffic through the campus but also due to the variation in the audio/video stream itself. This was observed while conducting the controlled experiment 6.3 in which the jitter and maximum jitter did not have consistent values even for a very sort duration of time. It kept changing as the audio/video progressed. Hence the campus traffic did not have much effect on the quality of the audio/video stream. And so the tunnel does not seem to have much overhead and the parameters of the multimedia stream is almost constant throughout the day.

The graph 6.5 gives a closer view on the variation of jitter throughout a day through the campus network.

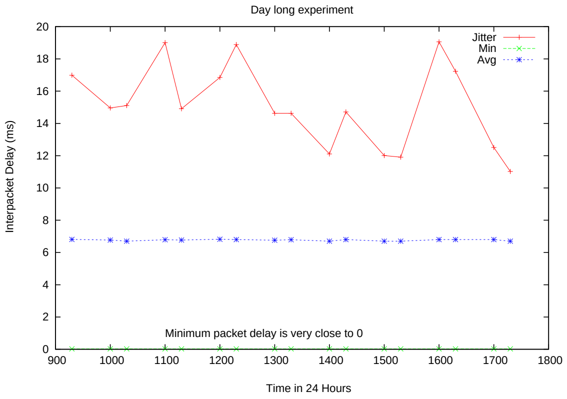


Figure 6.5: Jitter variation for a day.

The graph 6.6 shows the ping delay through the campus network (by tunneling) to a audio/video streaming server in the EDUSAT satellite network for a day. The graph shows the ping delay from the campus network through the tunnel we setup to a multicasting server in the EDUSAT satellite network. This valus was approximately in between 673 ms to 682 ms. But the variation in this value too was not due to campus network alone but also because of the change in response time of the multicast server itself. This could be due to the change in load of the multicast server or due to change in the network traffic within the satellite network itself. This was verified by pinging the multicast server directly without campus network coming into picture. Still the variation in the ping delay was almost the same. Hence we conclude that ping delay is also constant throughout the day and campus network does not affect it much.

All these results suggest that the traffic in the campus network does not pose any significant effect on our application of the tunnel for audio/video transmission to the

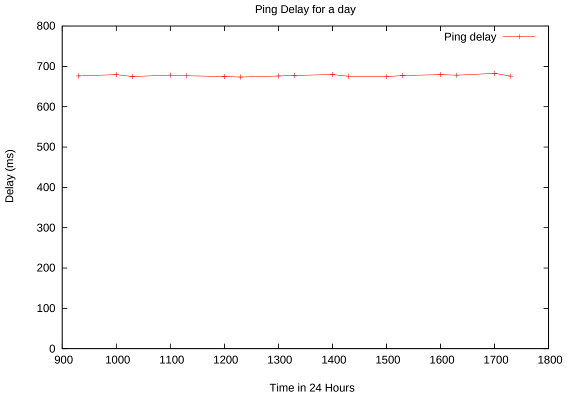


Figure 6.6: Ping delay for a day.

EDUSAT satellite network.

Further analysis of the performance of the tunnel through the campus network was done by pushing the limit on the bandwidth. For this a file was send between the tunnel gateways shown in Figure 5.1. The observation that was made here was that even this did not have any considerable impact on the stream parameters that was measured or even on the ping delay. The extra delay induced by overloading the bandwidth was about 10 ms. This is due to the fact that the bandwidth requirement for our application used for transmitting and receiving the live courses is very less which atmost is approximately 4000 Kbps as compared to the available bandwidth of 100 Mbps. So the possibility of our application failing due to high network traffic in the campus is still reduced. To verify this, we tried to push the network bandwidth usage by the applications used by CDEEP for transmitting the courses. This was done by increasing the bitrate of the audio/video stream. We were able to increase the bandwidth usage from 0.4% to about 2% of the available 100 Mbps ethernet network connection. But this too did not have much impact on the variation of stream parameters that were measured for analyzing the performance of the tunnel even after using the complete bandwidth between the tunnel gateways by a file tranfer.

Chapter 7

Challenges faced while Implementing and Testing

These are some of the challenges that were faced during the quest for a solution.

7.1 Fetching Network Details of CEEP EDUSAT

Getting the details of the network setup of CDEEP was not so obvious. So fetching the details about the configuration of computer systems used took sometime. Especially to draw conclusion of how the satellite network would look like diagrammatically. This includes getting details the gateway, streaming servers in the satellite network. The IP addresses of the machines used by CDEEP was easy to find out. But to find out the IP addresses used by the servers used in the satellite network took some effort. We had to ask banglore call centre for this. But it doesn't mean that they would readily give away the details because it is working for ISRO and so they needed a written permission from ISRO. Also the exact port on which to send the audio and video stream had to be known. This was finally inferred from the application that the CDEEP already uses to transmit their courses.

7.2 Understanding the Problem

The initial problem was to understand the given problem well and to come out with the answer to what has to be done exactly. We know that we have to somehow forward the multicast packets into the campus network. But what methods could be tried out was a difficult question. So we started trying out whatever came into mind until we understand better.

7.3 Finding out when the CDEEP EDUSAT Network was idle

The biggest challenge was to get the permission for testing any implementation on the CDEEP EDUSAT satellite network. This is because it is a live network and has to be dealt with carefully as institutions across India use the satellite network. So finding the exact time when the network is free from transmission is bit tedious task.

Also it is the responsibility of the CDEEP staff to ensure that no disturbance arises due to my testing. So definitely I have to convince well about my implementation before any testing can be done.

7.4 Encoding Problem with Application Layer Gateway

Though the application was developed and tested in our own environment, when it came to testing on the actual EDUSAT satellite network, it does not work just like that. To find out why is a more tedious question. Assuming that we were sending to correct server and port, we could conclude that the bit rate, encoding used for transmission was the reason for this. Many different encodings were tried but we couldn't get it working.

7.5 Sharing Streaming Devices over Network

Though the softwares were available for sharing hardware devices that allow sharing of hardware devices over network. Most of the softwares that I got where for sharing USB devices and especially mass storage devices. Even though these softwares seemed to work for mass storage devices, it was not so friendly with video capture devices such as the webcam.

As mentioned earlier, it was possible to share the hardware of a webcam. The hardware appeared on the remote machine as if it were its own. I even installed the device drivers for it. But when it came to the actual capturing or controlling of the device, the device did not respond at all. May be it needs a custom driver for such applications.

7.6 Testing NAT on Satellite Network

NAT was deployed quite easily. We had to carefully write down all the changes, such as IP addresses, that we would make to test our setup. But after the deployment every thing did work fine except that the return video that comes to the teacher's terminal did not work. To troubleshoot this took a lot of effort. First assumption was that there was some problem with the NAT configuration that did not allow certain network packets to be translated. The network packet monitoring tools were used to find out where the problem was. Finally it was found the receiving port at the teacher's terminal was mis-configured somehow. Though I don't remember making this error myself, I definitely had to pay for my reputation among the CDEEP staff that I would cause no disturbance. From next time I had to struggle harder to get their consent for doing the testing.

7.7 Network Layer Gateway for Tunneling

First of all, finding the right way to do tunnel was the challenge. The search was for a very easily configurable tool. Each one had to be tested.

7.7.1 Packets getting dropped at Routers

A lot of time was spent on making the GRE tunnel work though it never did. GRE tunnel worked in the same subnet. But when it was tested for tunneling through the campus network it did not work. We tried with a lot of configuration changes but still there was no hope. The conclusion was that the gateways at KReSIT were dropping the packets as GRE tunneling is IP-tunnel. So the tunneling is done at kernel by loading a module into the kernel. So since the gateways drop packets according the packet filtering rules of cc (computer centre of IIT Bombay) we try to find the exact reason. Later we found that the GRE did not use a port for encapsulating and decapsulating as most tunneling applications would do. But it uses IP protocol number 47 for this. So just as ICMP are dropped at the routers, these packets were also dropped.

7.7.2 Multicast Routing Daemon had problems

Hunting for a right routing daemon was to done. Since we came across a paper that made such an attempt which discussed about using FreeBSD machines that used mrouted for routing multicast packets. We too tried the same setup.

First of all we tried mrouted daemon on our same operating system we use (Ubuntu Jaunty). Since this was not distributed along with linux distribution because of proprietary issues, we had to download and try to install. But it wouldn't install easily. The dependencies were an issue. Most of the dependencies were old libraries that were difficult to get in new versions of linux. But still failed to have successful installation.

So instead of wasting time on this, decision was made to try on FreeBSD as they still have the package for mrouted daemon included in their distributions. Installation of FreeBSD took considerable time first of to understand their partitioning tool (which was quite different from linux and windows). Finally we managed to have an installation of FreeBSD. But when we tried to install and run mrouted on the GRE tunnel created, it was not able to do so because by default the FreeBSD do not enable multicast routing in the kernel. So we had to recompile the kernel to support multicast routing. Once this was done, the mrouted application did run. But still multicasting through the tunnel was not happening. We had to drop the idea of FreeBSD because of all these difficulties and as we wanted an easily configurable tool which would be more reliable.

Finally we came across OpenVPN. OpenVPN as mentioned earlier has support for tunneling at both network layer as well as at ethernet layer. Learning how to configure was time consuming. It is quite a complex tool with many options. A lot of problems were faced in order to make this just run. It make a secure tunnel and hence the key generation and installation of the keys takes time to learn. In the beginning it just showed that the TLS key was not received. We had to discuss with online forums on OpenVPN to get this sorted out. Finally OpenVPN was working. Also multicasting through the tunnel created.

7.7.3 Routing Problem of Multicast through Tunnel

The next challenge was to route multicast through the tunnel. Routing cannot be done through static routes for multicasting. Once we have a proper tunnel working now we can try multicasting daemons on the tunnel. Since mrouted was difficult the next try was pimd, which is the current industry standard for mutlicast routing which implements Protocol Independent Multicast routing. This was available in linux distributions and is free software, so no proprietary issues. Though it took sometime to get this working it did work. But it did not work for all multicast streams. Also it takes sometime for the stream to be routed as it takes time to add the group information into its table.

7.7.4 Solution was Bridging

For the problem of routing, we thought of an easier way to get through that would be more simple and reliable and fast as well. This was to bridge the tunnel interface itself. As the tunnel was an Ethernet tunnel this could be done without any effort as bridging is done at Layer 2 or MAC Layer.

7.8 Moving Teacher's Terminal and Video Server from KReSIT

The combination of tunneling and bridging was to be finally tested on the CDEEP EDUSAT satellite network. Now the challenge was to test the setup on the campus network and on the terminal application machines (Teacher's terminal and Video server). For this we had to move them out of KReSIT. But it had to done when there is no class in the Seminar Hall of 3rd Floor KReSIT. Finally we waited for the Holi Holidays so we could find time to do the testing. Testing was first without the bridge. Static routes were set to forward the stream of Video server through the tunnel. To test if the video was being received, we had to go back to the KReSIT building to check it on the Student's terminal. And it was working well.

But to get return video which is a ip-multicast stream, we cannot set static routes. So we tried the bridge on tunnel. And then everything began working fine. Teacher's terminal also worked as though the tunnel setup was completely transparent to the terminal machines. There was no need to change even the IP addresses of the terminal machines. Finally our solution is ready to be deployed.

7.9 Day Long Experiment for Performance Evaluation

To measure the tunnel performance for a complete day we had to do it at the time when the course is being conducted contrary to previous requirement of satellite network being free. We came with the idea of testing the return video from the multicast server from the satellite network. The bridged tunnel was made from 4th Floor CDEEP office to Hostel-12. The day long experiment was done for every hour.

7.10 Synchronization of Bridging and Tunneling

The application for automating the tunnel setup for CDEEP EDUSAT satellite network was easy to be developed in Java Swing. But there was a problem that had to be solved. This was because we had to setup the tunnel as well as bridging on the gateways during system startup. Now the challenge is that the bridging had to be done after the virtual interface of the tunnel is created while startup. If the order is reversed, then the bridge would not be applied to the tunnel interface. So we cannot guarantee while system startup that the tunnel interface is ready. A scripting was made that would check if the tunnel is ready or not. But this makes the system hang at startup. So the alternative way was to setup a cron (synchronization) job to bridge the required interfaces after a specified interval. Doing this has the advantage that even if the tunnel was disconnected and connected after system startup, it would still be bridged as it does bridging after an interval. Hence the problem was solved.

Chapter 8

Conclusions and Future work

We have discussed various possible ways of trying out to find a very appropriate solution. From our experimentation and observations we have found that the most effective and transparent solution is to implement the gateways at the network layer. When we do it at network layer, we do not have to make any changes either on the physical level or on the application level. The tunneling is the best way of transferring multicast packets through a unicast network. This technique requires no additional hardwares and is very cheap.

Now that we have a working solution, the main consideration will have to be the fact that the solution has to be scalable. In future definitely the number of users receiving as well as the number of servers will increase. So we will have to test our solution for all the above mentioned performance parameters. If the solution is not upto the mark then some improvements in the design of the tunnel itself will have to be made.

Bibliography

[1] B. fenner. mrouted. (url:ftp://ftp.parc.xerox.com/pub/net-research/ipmulti/).

[2] Openvpn. (http://openvpn.net/).

[3] E. Amir, S. McCanne, and H. Zhang. An application level video gateway. In *The Third ACM International Multimedia Conference and Exhibition (MULTIMEDIA '95)*, pages 255–266, New Yprk, November 1996. ACM Press.

[4] OpenVPN Setup. (http://openvpn.net/index.php/open source/documentation/howto.html).

[5] Bridging in Ubuntu. *(https://help.ubuntu.com/community/NetworkConnectionBridge)*.

[6] Takeru Inoue and Ryosuke Kurebayashi. Formulation of tunneling impact on multicast efficiency. *IEICE - Trans. Inf. Syst.*, E89-D(2):687–699, 2006.

[7] Milan Nikolic, Dan Hoffman, and Ljiljana Trajkovic. Mbone webcast: Network setup and data collection, proc. seventh iasted international conference on internet and multimedia systems and applications (imsa 2003), honolulu, hi, aug. 2003, pp. 140-145.

[8] Peter Parnes, Kre Synnes, and Dick Schefstrm. mtunnel: A multicast tunneling system with a user based quality-of-service model. In *European Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services*, pages 87–96, 1997.

[9] Peter Parnes, Kre Synnes, and Dick Schefstrm. Lightweight application level multicast tunneling using mtunnel. *Computer Communication*, 21:1295–1301, 1998.

[10] Tunneling Protocols. *(http://en.wikipedia.org/wiki/Tunneling_protocol)*.

[11] Tarik Čičić, Haakon Bryhni, and Steinar Sørlie. Unicast extensions to IP multicast. In *Proceedings of the Protocols for Multimedia Systems PROMS'2000*, pages 60–69, Kraków, Poland, 2000. ISBN 83-88309-05-6.

[12] D. Waitzman, C. Partridge, and S. E. Deering. RFC 1075: Distance vector multicast routing protocol, November 1988.

[13] OpenVPN Wikipedia. *http://en.wikipedia.org/wiki/OpenVPN*.

Appendix A

Configuration of Computers used by the CDEEP

A.1 Teachers Terminal Machine

Processor : Intel Pentium 4 2.79 Ghz
Graphics: NVIDIA GeForce 6200 Turbo Cache
RAM : 512 MB
HDD : 40 GB
Ethernet: Broadcom NetXtreme Gigabit Ethernet 10/100 Mbps
Operating System: Windows XP Professional SP2

A.2 Video Server Machine

Processor : Intel Core2Duo E 6550 2.33 Ghz
Graphics: Intel G33/G31 Express Chipset family
RAM : 2 GB
HDD : 300 GB
Ethernet: Gigabit Ethernet 10/100 Mbps
Operating System: Windows XP Professional SP2

A.3 Format of Video sent by the Video Server

This application gets the RAW audio/video feed directly from the hardware via standard windows driver. It then encodes and transmits the live audio/video to the multicast server in the EDUSAT satellite network. The application has many settings for the type of video that must be transmitted. Few parameters of the audio/video transmitted are:
VSS h.264 / AVC Codec
352 x 288 (12 bits) l420 Compression
Frame rate : 22 frames/second
Bit rate : 786 kbps

A.4 IP Addresses Allocation of CDEEP Machines

The IP addressed used in the machines running the Video Server application and the Teachers Terminal are in the range 10.254.75.x/24 where x is 2 for video server, 3 for Teachers terminal. The default gateway is 10.254.75.1. The IP addresses of the Students terminal at the remote center is in the range 10.254.85.x/24

Appendix B

Setting up the Tunnel using OpenVPN

Now we make a Ethernet-tunnel between PC-Server (openvpn server) and PC-Client (openvpn client).

B.1 On the PC-Server

copy the tools for creating keys to /etc/openvpn or any desired location of your choice

```
sudo cp -r /usr/share/doc/openvpn/examples/easy-rsa/2.0/ /etc/openvpn/easy-rsa
```

cd to the copied location

```
cd /etc/openvpn/easy-rsa
```

you will need root permissions to create and copy keys

```
sudo su
gedit vars
```

Change the values in quotes in these lines at the bottom so that they reflect your new CA.

Example:

```
export KEY_COUNTRY="IN"
export KEY_PROVINCE="MU"
export KEY_CITY="Mumbai"
export KEY_ORG="IIT-Bombay"
export KEY_EMAIL="tunnel@cdeep.iitb"
```

Enter these set of commands (you may refer to openvpn website [2] for what these commands do) from the same working directory

```
source ./vars
./clean-all
./build-dh
./pkitool initca
./pkitool --server server
./pkitool client
```

Now time to copy the created keys

```
cd keys/
```

copy server keys to the same machine's /etc/openvpn (default folder used by openvpn)

```
cp ca.crt server.crt server.key dh1024.pem /etc/openvpn
```

copy client keys to the remote machine using scp (or any other secure methods)

```
scp ca.crt client.crt client.key root@\$IP-Addr-Client\$:/etc/openvpn
```

Now you need to copy the sample server configuration file and edit it

```
cp /usr/share/doc/openvpn/examples/sample-config-files/server.conf.gz /etc/open
```

```
cd /etc/openvpn
```

extract the server.conf file

```
gzip -d server.conf.gz
```

edit the server.conf file

```
vim server.conf
```

you need to use tap virtual interface. This is because tap makes an Ethernet tunnel whereas tun makes an IP tunnel. For later bridging it, we need an Ethernet tunnel.

Change these:

```
;dev tap
dev tun
```

to (make use of tap device instead of tun):

```
dev tap
;dev tun
```

B.2 On the PC-Client

Copy and edit the client configuration file

```
sudo cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf /etc/op
sudo gedit client.conf
```

make same changes as in PC-Server for using tap interface

```
dev tap
;dev tun
```

Change my-server-1 to $IP-Addr-Server$ in the line so as to specify the server to connect:

```
remote $IP-Addr-Server$ 1194
```

You could disable the encryption feature of OpenVPN if you do not want the encryption and decryption overhead by adding the line

```
cipher none
```

to both client and server configuration files. This is completely optional.

B.3 Setting up the Bridge

We shall make a script to define bridge and make it start at boot. Do this on both the PC's.

```
sudo vim /etc/init.d/bridge-start
```

This is how the script should look like. Do necessary changes for it to refer to proper interfaces.

```
#!/bin/bash

#################################
# Set up Ethernet bridge on Linux
# Requires: bridge-utils
#################################

# Define Bridge Interface
```

```
br="br0"

# Define TAP interface to be bridged
tap="tap0"

# Define physical Ethernet interface to be bridged
# with TAP interface(s) above.
eth="eth1"

brctl addbr \$br
brctl addif \$br \$eth
brctl addif \$br \$tap
ifconfig \$eth 0.0.0.0
ifconfig \$tap 0.0.0.0
ifconfig br0 up
```

Here eth1 refers to the interface as mentioned in section Physical Setup. If it is some other interface then make necessary changes.

Now make the script executable and add it to startup

```
sudo chmod +x  /etc/init.d/bridge-start
sudo update-rc.d /etc/init.d/bridge-start defaults
```

Now restart both PC-Server and PC-Client. You should now be able to see tunnel interface, tap0, and bridge interface, br0, on both PC's.

This ends the setup and configuration. Now once you connect the end terminals that run applications for conducting live sessions (Teacher's terminal or Video server) to the eth1 interface on the PC-Client as shown in Figure 5.1, you should be able to transmit and receive traffic via the Edusat Satellite network including IP-mutlicast. You do not have to make any changes on the end terminals.