An Architecture for Multicasting using Private Tunnel

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in

Software Engineering

Submitted By

Amandeep Singh Sidhu

(Roll No. 801131002)

Under the supervision of:
Dr. Neeraj Kumar
Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

June 2013

Acknowledgement

My time as a postgraduate student at Thapar University, Patiala is supported by many kind people. This page attempts to recognize those who have been most helpful along the way.

First of all, I thank God for providing me such an opportunity and support.

I am deeply indebted to my supervisor, Dr. Neeraj Kumar. His advice, support and kind encouragement on thesis and professional issues have effectively guided me to the right path. He always led my research papers to clear, smart and effective ones. Without him this thesis would not be possible.

Thanks to all members of Thapar University. Dr Maninder Singh, Head of Computer Science Department, for his kind support and directing me in field of research based on his extensive knowledge. All staff members of Computer Science and Engineering Department for providing me all the resources required for the completion of my thesis.

I would like to thank my friends, Rajinder Singh Sidhu and Amritpal Singh for their moral and technical support. They always encourage me when I needed the most.

Outside the Computer Science and Engineering Department, I benefited from time spent with friends especially Atul Sharma , Mandeep Singh and Gurusharan Virk. Interdisciplinary discussion with these friends has given me many new ideas and perspectives to work on my thesis.

I want to express my appreciation to each and every person who directly or indirectly help me to complete my thesis. Finally, I thank my family for all aspects of life.

Amandeep Singh Sidhu

Certificate

I hereby certify that the work which is being presented in the thesis entitled, "*An Architecture for Multicasting using Private Tunnel*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Neeraj Kumar* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Amandeep

(Amandeep Singh Sidhu)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Dr. Neeraj Kumar)

Assistant Professor

Computer Science and Engineering Department

Countersigned by

(Dr. Maninder Singh)

Head

Computer Science and Engineering Department

Thapar University

Patiala

(Dr. S. K. Mohapatra)

Dean (Academic Affairs)

Thapar University

Patiala

Abstract

Multicast IP is a routing scheme for efficient streaming of media and other content over a network to selected host. By multiplication of the data in the routers on the way to their destination, it allows servers to save bandwidth by only sending one stream to multiple receivers. Using IP multicast techniques to distribute this information can often substantially reduce the overall bandwidth demands on the network. A good example of this approach is the rapidly growing area of audio and video Web content.

The use of multicast over the internet is rather limited as most of the ISP router does not support Multicasting for lack of hardware support. In this thesis multicast will be explained and a simple method of sending multicast data over internet to other networks.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

Internet started to exchange information mainly by text and data, but it has evolved as a stream of digital information of any form. We can reach data as in software, news and email as text, collected data in databases, graphical images, music, video, and many more. It is clear that all these forms of data are different and they might need a different approach to handle.

Streaming can be done several ways if you talk about the use of the network. The three main ways to stream content is through unicast, multicast or broadcast. They all have their advantages and disadvantages. However not all of them are supported by the internet backbone and the Internet Service Providers. Today most of the streaming is done by a unicast connection which make a dedicated own connection. This is suitable for on-demand streaming as in case of youtube, as users are streaming stored content on a server as they were downloading a simple file.

However when there is live content viewers access the same source and same content simultaneously. This means that the same live stream data have to be send simultaneously to different receivers. With a unicast system all those users would need a separate connection and all the load would be on that one server. This is not efficient as all are requesting same content and there where be multiple copy of same data and therefore there are the two other techniques: multicast and broadcast. They use the network structure itself to duplicate or multiply the live stream to the receivers, and take away the load from the server. The technique is rather simple but yet not supported or even blocked by Internet Service Providers and the internet backbone.

It is routing that makes the internet work today. Routing decides which data goes where. To manage all this routers have routing protocols and routing schemes. Protocols are used to format data, get network topology and agree on ways of communications. Routing schemes are used to define a method for the data to be delivered to the client device, to decide how the data has to get to the destination.

There are a couple of different types of devices in the topology of a network. The devices that deliver the content to the network are called the servers. They store the websites,

data, multimedia and all other content delivered on the internet. On the other end of the line there are the clients. These are the computers that browse the websites, smart phones that display a video from YouTube or an interactive television. In between of the clients and servers we have the network devices, mainly routers, firewalls and switches. They do everything to deliver the data, called packets, sent from the servers to the clients. Firewalls control the safety and block unwanted packets. Routers route the packets to the right network and switches deliver it to the destined client, router, switch, firewall or server. There are also devices called hubs, they send all packets to all their ports. This chapter provides a brief overview of Routing schemas by discussing its types and different application areas. Motivation and organization of thesis are discussed at the end of this chapter.

1.1 Unicast

Unicast is the most general way of sending packets through a network. 'Uni' stands for 'one' or 'single'. It means in short that in unicast there is build a single cast or 'stream' of packets from the server to each client. It is a one to one relation. In Figure 1.1 you can see a symbolic representation of unicast.

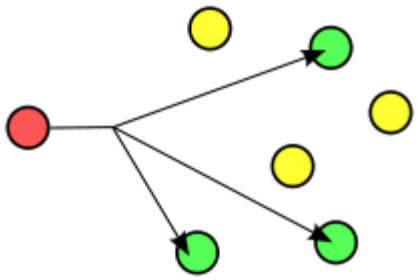


Figure 1.1 Unicast Network

Unicast is the standard method for transferring packets on a network. The most common services of the internet use unicast. Connecting to a web server to get a webpage over http, downloading files of an ftp server, VOIP conversations with Skype, streaming a video on YouTube, sharing files over the Bit Torrent network, sending an email with smtp or a connection through telnet, it are all example of unicast [1]. Unicast uses TCP and UDP to deliver the packets.

An IP internetwork, like any computer internetwork, consists of smaller networks joined together by the use of interconnecting devices known as routers. IP routing is the process of forwarding IP packets from a network device on one part of an IP internetwork to a network device on another network segment (subnet). An IP packet or datagram is a unit of information sent over an IP network that includes data intended for the recipient as well as a header containing routing information (the source and destination addresses and error-control data). IP routers forward packets between network segments.

Unicast IP routing is the process that enables unicast IP packets to be forwarded across an IP internetwork from a sending node to a destination node through one or more intermediate routers. A node is any network device that is running the TCP/IP protocol. A host is a node that does not perform routing, such as a user workstation or a non-router server. A router is a node that performs routing. That is, a router forwards packets that are not destined for the router itself either directly to the destination or to another router on the route to the destination.

Over the past few decades, the ability for people to communicate by sending messages from one computer to another, whether the computers are located on the same network in an office building or on networks at opposite sides of the globe, has become so commonplace that it is difficult to imagine a world where such communication is not possible. IP routing provides the infrastructure that enables all other IP-based technologies, such as Domain Name System (DNS), to function. The routing of unicast IP packets over IP internetworks is a major part of the technology that makes such communication possible.

Today, the majority of internetwork traffic worldwide is over IPv4 networks, and most user-initiated traffic across IPv4 internetworks is unicast traffic. Unicast IP routing occurs on every IP internetwork, including:

- An IP intranet not connected to the Internet
- The global Internet
- Intranets that connect to the Internet or to each other through the Internet

The major operating systems for which TCP/IP is the primary network protocol are Windows and UNIX.

Any Windows network supports unicast IP routing. These include networks that use only hardware routers, networks that use software-based routers such as the Routing and Remote Access service included with Windows Server 2003, or networks that use a combination of hardware and software routers.

Although all modern networking operating systems support the TCP/IP protocol suite, Windows Server 2003 TCP/IP provides the best platform for connecting Windows-based systems to earlier Windows-based systems and to non-Windows-based systems. In Windows Server 2003, TCP/IP supports enterprise networking and connectivity on computers running the Windows Server 2003, Microsoft Windows XP, Windows 2000, Windows NT, Windows Millennium Edition, and Windows 98 operating systems and on computers running the UNIX operating system.

1.2 Broadcast

Broadcast is somewhat the opposite of unicast. In broadcast packets are send to everyone on the network. All clients receive the packets and have to decide for themselves if the packets are interesting or not. It is a one to many relation, in Figure 1.2 you can see a symbolic representation of broadcast.

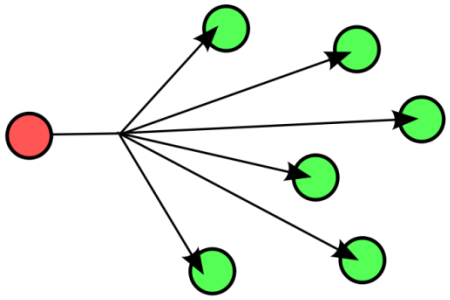


Figure 1.2 Broadcast network

A typical example that uses broadcast is Address Resolution Protocol (ARP). It uses broadcast to send the ARP packet to all computers on the LAN [1].

For broadcasting a host will send packets to the network broadcast address. The broadcast address of a network is the address where all the bits of the host part of the

IP address are set to one. For example the network 192.168.1.0/24 its broadcast address is 192.168.1.255. If the host does not know the current network, like a new client in a DHCP environment, it will broadcast to the standard physical network broadcast address, where all the bits of the IP address are set to one, so for IPv4 it gives 255.255.255.255. Routers will never forward this packets out of a network.

In it most pure form, a router will redistribute a broadcast packet to all connected networks. However this might cause packets to get into a loop (for example with redundant networks) until the Time To Live (TTL) has reached a value of zero. A subset of broadcasting is multidestination routing. With multidestination routing each packet contains a list of receivers. This way a router checks to which networks the packets have to be send, and they do not cause unnecessary flooding of the network.

There is however a more efficient way of broadcasting, called reverse path forwarding. With reverse path forwarding a router checks each source of each incoming broadcast packet. When the link where the router got this packet from is the most optimal link to get to its source, the router assumes it's the first packet to arrive, and will broadcast it to all other connections, except for the incoming link. However if the link where the broadcast packet arrived is not the most optimal path to the source of the packet, the router will discard this packet, as it is probably a double packet. To use this form of broadcasting, the routers only need to know the most optimal path, they are able to get this information either with distance vector routing or with link state routing. An even more efficient way is the use of a spanning tree. With a spanning tree the router knows the topology of the network, so when a broadcast packet arrives, it knows to which networks it should distribute the packets itself. This however requires link state routing

Broadcasting is by nature a connectionless routing scheme. This is because there is no real connection established between source and destination. Therefore broadcast can only use the UDP and not the TCP.

1.3 Multicast

Multicast is a more efficient way of broadcast. Unlike broadcast it will only send the packages to the clients that are in the multicast group. It is like broadcast a one to many relation. In Figure 1.3 you can see a symbolic representation of multicast.

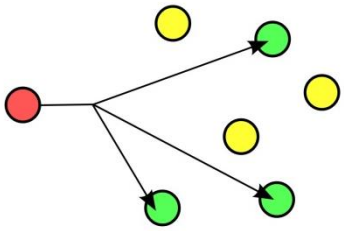


Figure 1.3 Multicast Network

IPTV is one of the applications that uses multicast to make efficient usage of bandwidth on the network.

Multicast packets are sent to a multicast IP address. For each multicast group there is such an IP address, and computers who want to join the multicast group send and listen also on this IP addresses. These IP addresses are assigned directly by IANA, thus they are reserved. In Table 1.1 [2] the ranges of the multicast addresses are given.

Table 1.1 Range of the multicast

IP multicast address	**Description**
224.0.0.0	Base address (reserved)
224.0.0.1	The *All Hosts* multicast group addresses all hosts on the same network segment.
224.0.0.2	The *All Routers* multicast group addresses all routers on the same network segment.
224.0.0.4	This address is used in the Distance Vector Multicast Routing Protocol (DVMRP) to address multicast routers.
224.0.0.5	The Open Shortest Path First (OSPF) *All OSPF Routers* address is used to send Hello packets to all OSPF routers on a network segment.
224.0.0.6	The OSPF *All Designated Routers* ""(DR)"" address is used to send OSPF routing information to designated routers on a network segment.
224.0.0.9	The Routing Information Protocol (RIP) version 2 group address is used to send routing information to all RIP2-aware routers on a

	network segment.
224.0.0.10	The Enhanced Interior Gateway Routing Protocol (EIGRP) group address is used to send routing information to all EIGRP routers on a network segment.
224.0.0.13	Protocol Independent Multicast (PIM) Version 2
224.0.0.18	Virtual Router Redundancy Protocol (VRRP)
224.0.0.19 - 21	IS-IS over IP
224.0.0.22	Internet Group Management Protocol (IGMP) Version 3
224.0.0.102	Hot Standby Router Protocol version 2 (HSRPv2) / Gateway Load Balancing Protocol (GLBP)
224.0.0.107	Precision Time Protocol version 2 peer delay measurement messaging
224.0.0.251	Multicast DNS (mDNS) address
224.0.0.252	Link-local Multicast Name Resolution (LLMNR) address
224.0.1.1	Network Time Protocol clients listen on this address for protocol messages when operating in multicast mode.
224.0.1.39	The Cisco multicast router *AUTO-RP-ANNOUNCE* address is used by RP mapping agents to listen for candidate announcements.
224.0.1.40	The Cisco multicast router *AUTO-RP-DISCOVERY* address is the destination address for messages from the RP mapping agent to discover candidates.
224.0.1.41	H.323 Gatekeeper discovery address
224.0.1.129 – 132	Precision Time Protocol version 1 time announcements
224.0.1.129	Precision Time Protocol version 2 time announcements

1.4 Anycast

Anycast is neither a real broadcast nor a real unicast routing scheme. With anycast packets will be send to one receiver that is available in a group of receivers. It is a one to one-of-many relation. In Figure 1.4 you can see a symbolic representation of anycast.

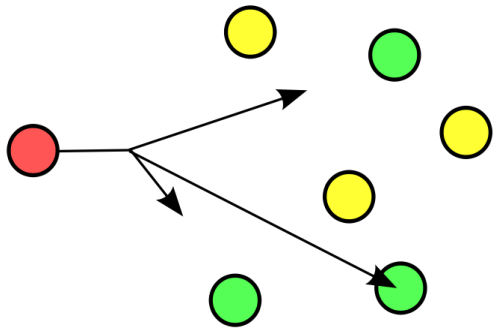


Figure 1.4 Anycast

An example of anycast is root DNS servers, to provide them with redundancy. If one of them is not available the packet will be resend to the next address in the pool of receivers. This provides a continuous availability of the DNS service that is needed for every single lookup of a domain name

Anycast is mainly intended to use UDP but these days also more and more TCP applications are used, like content delivery services [3].

1.5 Geocast

The last routing scheme is geocast. As the name itself implies, it uses geographical conditions to deliver the packets, as sending packets to all receivers in one specific location. Geocast is a one to many relation, which makes it similar to broadcast and multicast. In Figure 1.5 you can see a symbolic representation of geocast.

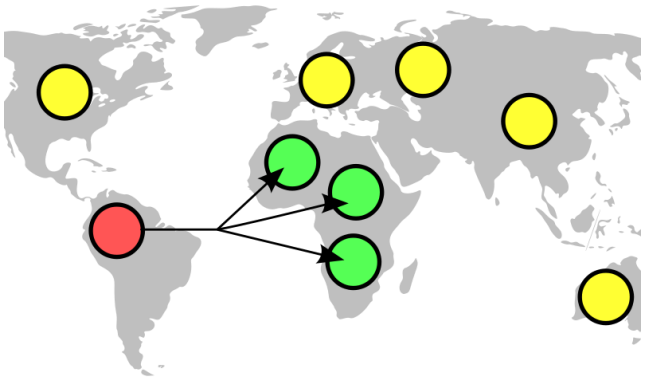


Figure 1.5 Geocast Network

Until today there are not any significant applications that use the geocast routing scheme. In the nineties geocast was stated as a promising routing scheme. They said it has many opportunities, for commercial and governmental use [4].

1.6 Research Motivation

Despite the advancement in Multicast from industry and academics contributions, Multicast faces many challenges for its worldwide adoption. The primary reason for not adopting the Multicasting is that many ISP does not support multicasting due to its increase in hardware cost. This locks multicast user to local network infrastructure. If all the ISP make their network multicast compatible on widely adopt and promote making it more easier to implement.

Internet compatibility is missing, there are number of overlay networks available for multicast but they have there limitation for i.e. The MBONE was designed to use 500 Kbps to transmit multicasts, which is an adequate maximum bandwidth for now [5].

Using simple and secure method of sending multicast data as unicast to cross non multicast support router. Aim of this thesis is to create simple network architecture with the use existing tools to send multicast data over internet to various users.

1.7 Organization of Thesis

Rest of this thesis is organized as follows:

Chapter 2 – This Chapter summarizes literature survey done to study the concept of Multicasting over internet, Load on multicasting router and application use of multicast.

Chapter 3 – This chapter focuses on research gaps and also state problem statement for this thesis.

Chapter 4 – This chapter provides solution to problem stated in previous chapter by proposing architecture and explaining its all components. Designing of system using Network diagram is also explained.

Chapter 5 – This chapter demonstrate all parts of proposed architecture by tools that can help us achieve the proposed architecture and demonstrate using Bit torrent and putty etc. tools. Results of experimental setup are shown.

Chapter 6 – This chapter describes the conclusion, contribution of work and future research possible.

Chapter 2

Literature Review

In the previous Chapter, different Routing Schemas is introduced by discussing it's their delivery mechanism. Now discuss one of crucial open challenges in field of Multicasting is streaming over internet. Focusing on stated challenge this chapter discuss state of research work in the field of streaming of data using Multicasting IP. This literature review explores different ways of sending multicast data over internet. For multicast to work all over the internet it is required for every device, or at least the local nodes network to support multicast. It is not a question if the devices are capable or not, because they are, but more a question if it is favourable to activate the support for multicast. A lot of ISP's for example charge users by the amount of data they download, or give users a monthly bandwidth. With multicast they would not be able to see all the traffic passing through per user. Also content providers like to know how much users are watching their content. As it is much easier with unicast to count the users, which are needed for commercial income, they do not tend to use multicast over unicast. The cost of providing unicast does not weigh against the revenues of statistics which are valuable in the commercial world. That being said it does not mean multicast over the internet is completely impossible. We will discuss multicast data ways of multicasting data over internet.

2.1 Mbone

Mbone is a virtual network built on top of the Internet, invented by Van Jacobson, Steve Deering and Stephen Casner in 1992. The purpose of Mbone is to minimize the amount of data required for multipoint audio/video-conferencing. Mbone is free; it uses a network of mrouters that can support IP multicast, and it enables access to real-time interactive multimedia on the Internet. Many older routers do not support IP multicast. To cope with this tunnels must be set up on both ends; also known as a tunneling protocol: multicast packets are encapsulated in unicast packets and sent through a tunnel. Mbone uses a small subset of the class D IP address space (224.0.0.0–239.255.255.255) assigned for multicast traffic. Mbone uses 224.2.0.0 for multimedia conferencing [6].

2.1.1 Application of Mbone

- Virtual network created by chaining machines called mrouters.
- This network of "nodes" works around the router barrier problem by sending packets to the next mrouter using a regular machine to machine unicast.
- Send data packets to multiple sites at the same time.
- Avoids sending duplicate data to over the network by sending the same packet to any number of machines.
- Does not normally send information off of a local network.

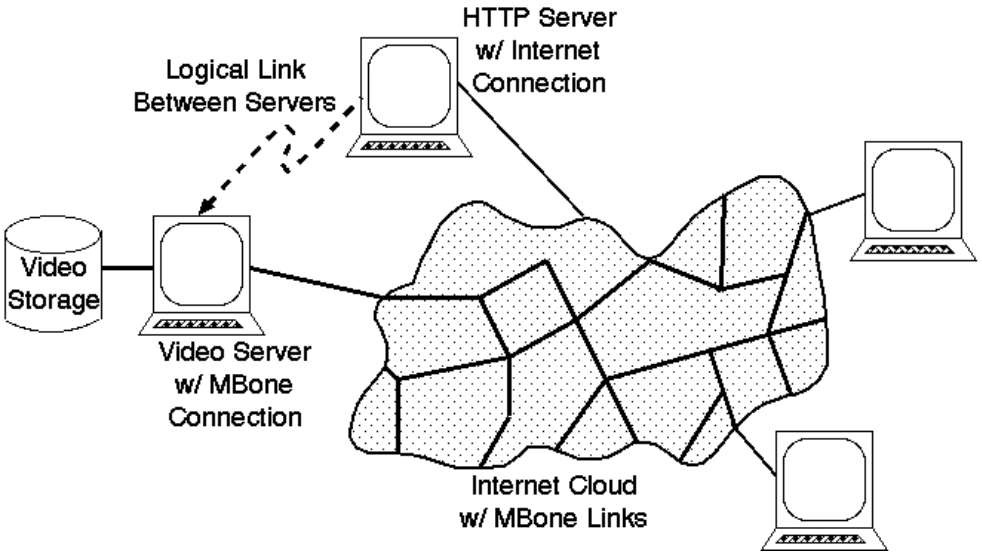
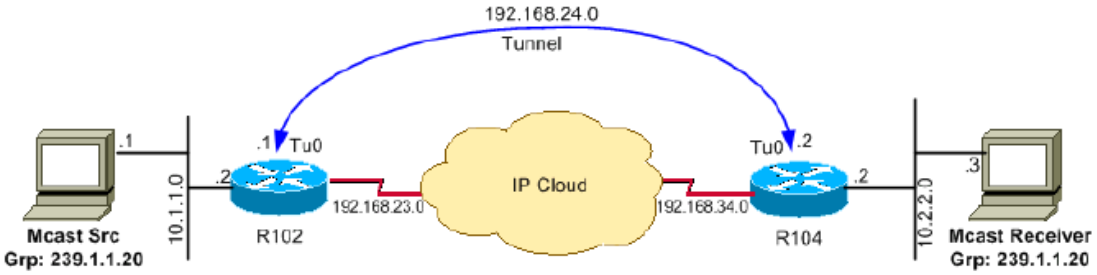


Figure 2.1 Mbone Network

2.2 Mcast

Mcast was a project from the Vrije Universiteit Brussel trying to enable multicast access for everyone on the internet. With tunnelling technology they tried to capsulate the packets over the existing network to other multicast nodes. The configuration of the tunnel clients running on the client computers would be automatic.

The researchers hope was to show ISP's and content providers the usefulness of multicast. If enough clients would be using multicast, they could be convinced of investigating in multicast on the ISP's network. However after the project professor Marnix Goossens said they under estimated the current business model. As for now t



here is nobody at the Vrije Universiteit Brussel researching multicast anymore [7].

2.3 6Bone

The 6bone is an independent outgrowth of the IETF IPng project that resulted in the creation of the IPv6 protocols intended eventually to replace the current Internet network layer protocols known as IPv4. The 6bone is currently an informal collaborative project covering North America, Europe, and Japan.

One essential part in the IPv4 to IPv6 transition is the development of an Internet-wide IPv6 backbone infrastructure that can transport IPv6 packets. As with the existing IPv4 Internet backbone, the IPv6 backbone infrastructure will be composed of many Internet Service Providers (ISPs) and user networks linked together to provide the world-wide Internet.

Until the IPv6 protocols are widely implemented and fully tested for interoperability, production ISP and user network routers will not readily place production Internet (IPv4) routers at risk. Thus a way is needed to provide Internet-wide IPv6 transport in an organized and orderly way for early testing and early use.

The 6bone is a virtual network layered on top of portions of the physical IPv4-based Internet to support routing of IPv6 packets, as that function has not yet been integrated into many production routers. The network is composed of islands that can directly support IPv6 packets, linked by virtual point-to-point links called "tunnels". The tunnel endpoints are typically workstation-class machines having operating system support for Ipv6.

Over time, as confidence builds to allow production routers to carry native IPv6 packets, it is expected that the 6bone would disappear by agreement of all parties. It would be replaced in a transparent way by production ISP and user network IPv6 Internet-wide transport [8].

The 6bone is thus focused on providing the early policy and procedures necessary to provide IPv6 transport in a reasonable fashion so testing and experience can be carried out. It would not attempt to provide new network interconnect architectures, procedures and policies that are clearly the purview of ISP and user network operators. In fact, it is the desire to include as many ISP and user network operators in the 6bone process as possible to guarantee a seamless transition to IPv6.

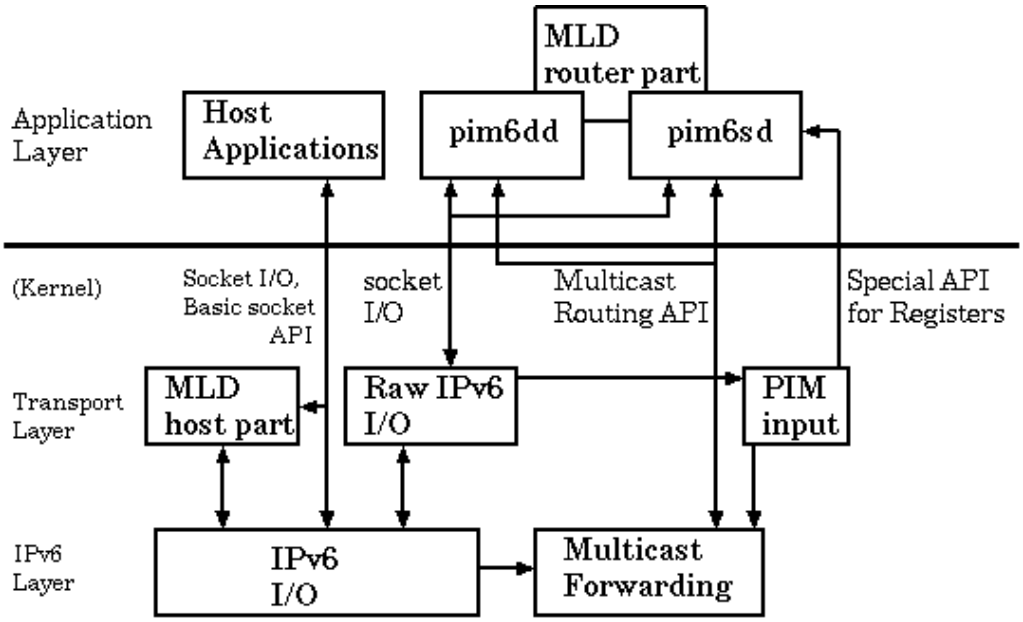


Figure 2.2 6Bone Network [8]

2.4 AMT—A Transparent Bridge to Full IP Multicasting

Automatic Multicast Tunneling (AMT) technology in its Juniper Networks® Junos®operating system for MX Series 3D Universal Edge Routers overcomes the lack of multicast capability on a given device. If lack of multicast support on any device along the path prevents a particular user from joining a native multicast stream, then an AMT gateway which is built into Octoshape's powerful and resilient video distribution technology requests that an AMT - enabled router join the multicast on behalf of the user. Specifically, AMT establishes tunnels that link users on unicast-only networks with the content they want on multicast enabled networks. If you operate a multicast-enabled network, you can deploy AMT relays, i.e., AMT-enabled routers, at the edge of the network. As an AMT tunnel endpoint, the AMT relay essentially "translates" native IP multicast content for users on a unicast only network. Users on the unicast-only network have an AMT gateway—the other tunnel

endpoint—on their devices. This gateway,i.e., Octoshape's resilient video distribution technology, uses the "anycast" autodiscovery mechanism to locate the nearest AMT relay and then dynamically initiates an IP multicast tunnel to that relay [9].

The AMT gateway asks the AMT relay to forward a multicast content stream through the tunnel. Upon receiving the request, the AMT relay joins the multicast content, and the multicast stream is forwarded through the multicast enabled network to the AMT relay which, in turn, forwards it via the tunnel to the user's AMT gateway. Thanks to AMT technology, everyone on the network can now receive the signal via multicast, even if multicast is not supported in every part of the network, all the way down to the user. By serving as a transparent bridge that allows users to "hop over" unicast-only networks, AMT resolves the last-mile challenges facing:

• Service providers, who want the maximum return on their capital expenditure investments.

• Content providers, who want to minimize the cost they incur for delivery of their content.

• Users, who want access to any content, regardless of whether someone delivers that content over a unicast or a multicast infrastructure.

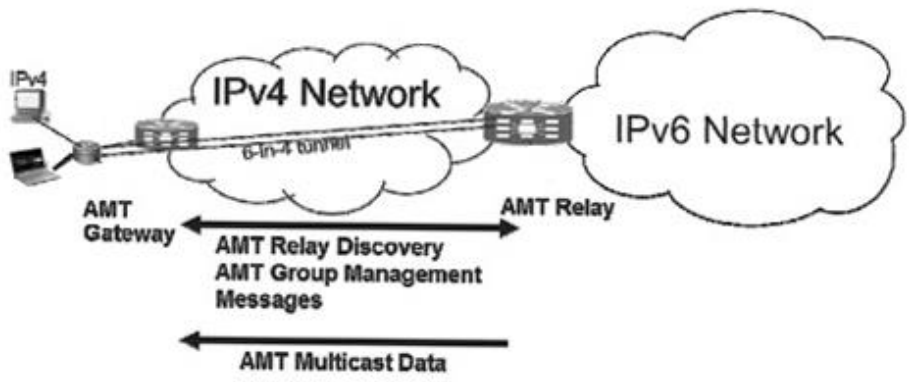


Figure 2.3 AMT Network [9]

2.5 Conclusion

In this chapter, methods of streaming multicast over internet are discussed. All the system requires 3rd party network or special hardware to implement multicast. After a

literature review is done for multicasting using overlay network. Based on research gaps found in this literature review next chapter formulates the problem statement.

Chapter 3

Gap Analysis and Problem Statement

Based on literature survey in the previous chapter, gaps have been identified in intra domain multicast. Gaps identified in this chapter will be discussed. Based on these gaps we formulated problem statement for this thesis in next section. Objectives of this thesis are enlisted at end of this chapter.

3.1 Gap Analysis

As in previous chapter, literature survey is done on various methods of sending multicast data over internet.

3.1.1 Complex System

All the system that has been developed for multicasting are complex to implement. As they require high level of knowledge how to deploy them. If you want multicast data for small amount of time it is not beneficial.

3.1.2 Third Party Networks

Multicasting will depend on 3rd party networks, if that network is down multicasting will not work. Which have their own limitation for example Mbone support only 500kbps of bandwidth [7]. If 3rd party network fails then your multicast network will too.

3.1.3 Reliability

Most of these logical networks running on top of IP layer are based on RTP (Real time protocol) that does not support reliability. Application running on these networks often experience poor quality due to packet lost as network configuration as per application is not present.

3.1.4 Commercial System

There is no commercial network that is providing multicasting over internet, all the overlay network that are running now are not developed or maintained aggressively. All are under development and evolving every day.

3.2 Problem Statement

This thesis addresses four different issues in Multicasting in recent times.

I. Multicasting data should be made easier to send over network with present system it is not easy to do as they have own methods or standards that one should follow.
II. Creating own network to multicast should be setup so that we can manage and control it.
III. Depending upon type of data we are sending we can have option to choose what protocol should we used is data we sending time critical or not.
IV. We should we able to use data with commercial system if any reliable network is there.

In this section four different issues in Multicasting are discussed. A solution to each issue is also provided. Solving these problems in a single architecture is what Multicasting needs now. Next sections states objectives of this thesis so that above issues can be resolved.

3.3 Objectives

Some of objectives of this thesis are:

1. Working on new method to multicast over internet. Objective is to create simple and easily overlay network for multicast.
2. Get multicasting data from other network with minimum hardware.
3. Network should be secure over internet.

Chapter 4

Proposed Solution

Studying the gaps and setting the objectives of this thesis in previous chapter. In this chapter we propose a solution to setup environment and resolve some of issues. Requirement analysis is done based on objectives of this thesis. Lastly we propose architecture for multicasting data over internet. Each component of architecture is explained in detail.

4.1 Requirement Analysis

In this section requirements are collected based on which architecture will be developed. Main focus to propose solution is to create simple multicast without 3rd party solution. These four layers are:

I. **Streaming Server**: Streaming servers typically deliver files to you with or without a help from a Web server. First, you go to a Web page, which is stored on the Web server. You might be running some software application like VLC player that can stream content directly. When you click the file you want to use, the Web server sends a message to the streaming server, telling it which file you want. The streaming server sends the file directly to you, bypassing the Web server.

   All of this data gets to where it needs to go because of sets of rules known as protocols, which govern the way data travels from one device to another.

II. **Client Application**: Application that will help us to request and receive the required multicast information from the stream server.

III. **Mrouters:** An mrouter can disguise multicast packets so that they can cross unicast routers. This is done by making each multicast packet look like a unicast packet, the destination address is the next mrouter.

When these three layers are linked through network for information exchange many issues can be resolved. For example, streaming data is multicast over internet.

4.2 Receiving Multicast Datagrams

Multicast follow IGMP request to join network IGMP is an integral part of IP multicast. IGMP can be used for one-to-many networking applications such as online streaming video and gaming, and allows more efficient use of resources when supporting these types of applications [10].

4.2.1 Joining a Multicast Group

Broadcast is (in comparison) easier to implement than multicast. It doesn't require processes to give the kernel some rules regarding what to do with broadcast packets. The kernel just knows what to do read and deliver all of them to the proper applications [11].

With multicast, however, it is necessary to advise the kernel which multicast groups we are interested in. That is, we have to ask the kernel to "join" those multicast groups. Depending on the underlying hardware, multicast datagrams are filtered by the hardware or by the IP layer (and, in some cases, by both). Only those with a destination group previously registered via a join are accepted.

Essentially, when we join a group we are telling the kernel. By default, you ignore multicast datagrams, but remember that I am interested in this multicast group. So, do read and deliver (to any process interested in them, not only to me) any datagram that you see in this network interface with this multicast group in its destination field.

Some considerations: first, note that you don't just join a group. You join a group on a particular network interface. Of course, it is possible to join the same group on more than one interface. If you don't specify a concrete interface, then the kernel will choose it based on its routing tables when datagrams are to be sent. It is also possible that more than one process joins the same multicast group on the same interface. They will all receive the datagrams sent to that group via that interface.

Any multicast-capable hosts join the *all-hosts* group at start-up , so "pinging" 224.0.0.1 returns all hosts in the network that have multicast enabled.

Finally, consider that for a process to receive multicast datagrams it has to ask the kernel to join the group and bind the port those datagrams were being sent to. The

UDP layer uses both the destination address and port to demultiplex the packets and decide which socket(s) deliver them to.

4.2.2 Leaving a Multicast Group

When a process is no longer interested in a multicast group, it informs the kernel that it wants to leave that group. It is important to understand that this doesn't mean that the kernel will no longer accept multicast datagrams destined to that multicast group. It will still do so if there are more processes who issued a "multicast join" petition for that group and are still interested. In that case the host remains member of the group, until all the processes decide to leave the group [12].

Even more: if you leave the group, but remain bound to the port you were receiving the multicast traffic on, and there are more processes that joined the group, you will still receive the multicast transmissions.

The idea is that joining a multicast group only tells the IP and data link layer (which in some cases explicitly tells the hardware) to accept multicast datagrams destined to that group. It is not a per-process membership, but a per-host membership.

4.2 Design of the Network Architecture

The Network Design is to understand how network architecture is going to be established. Help us understand different parts of the network. It is expressive way of addressing all the needed to deploy the system. With the help of network design tools we will propose the network architecture for the implementing multicasting application. In this we will create architectures to send multicast data to multiple receivers over internet.

4.2.1 Multicast data over internet with blocked ISP

We will start with the diagram to give overview of the multicast data streaming over internet.

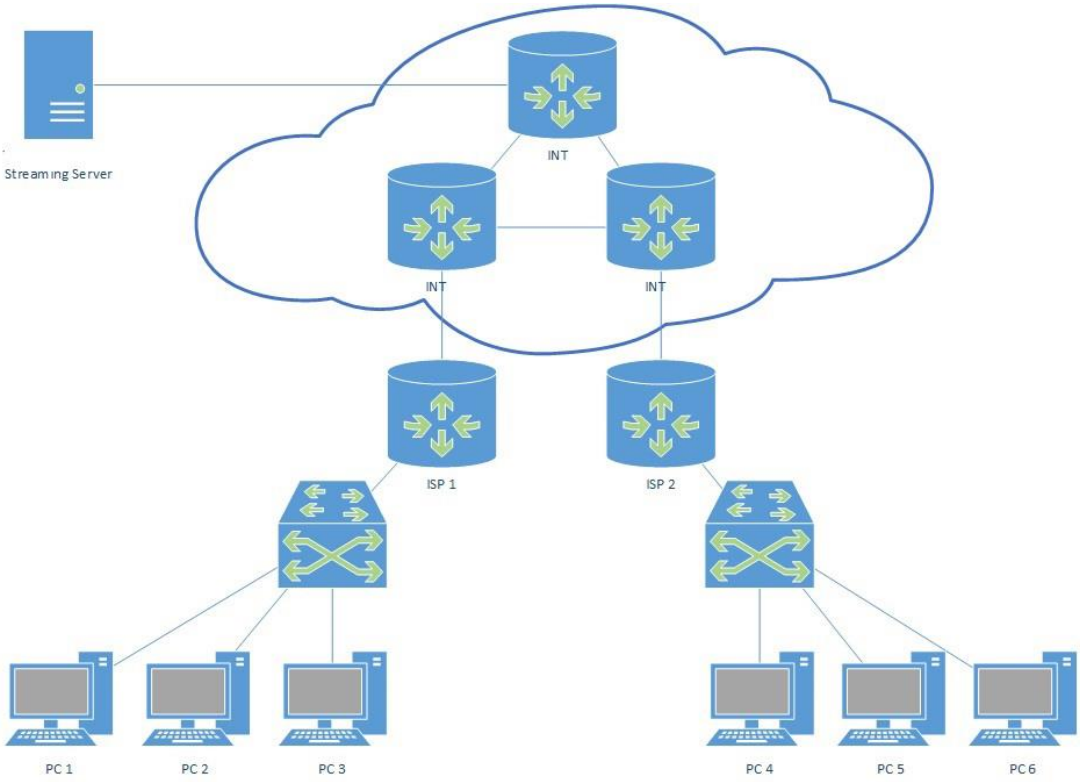


Figure 4.1 General overview of multicasting over internet

In figure 4.1 streaming server will send content to the requested clients PC 1 to PC 6, there are connected by two different ISP. ISP 1 router supports multicast routing and ISP 2 does not support. So when media server will stream data PC 1 to PC 3 will receive the requested data which can be audio, video or any other kind of multicast data. But PC 4 to PC 6 will not receive any data as they are connected through ISP 2.

4.2.2 Multicast data through blocked using XORP

We will create architecture to multicast data from streaming server to other network using tunnelling process with the help of XORP mroute.

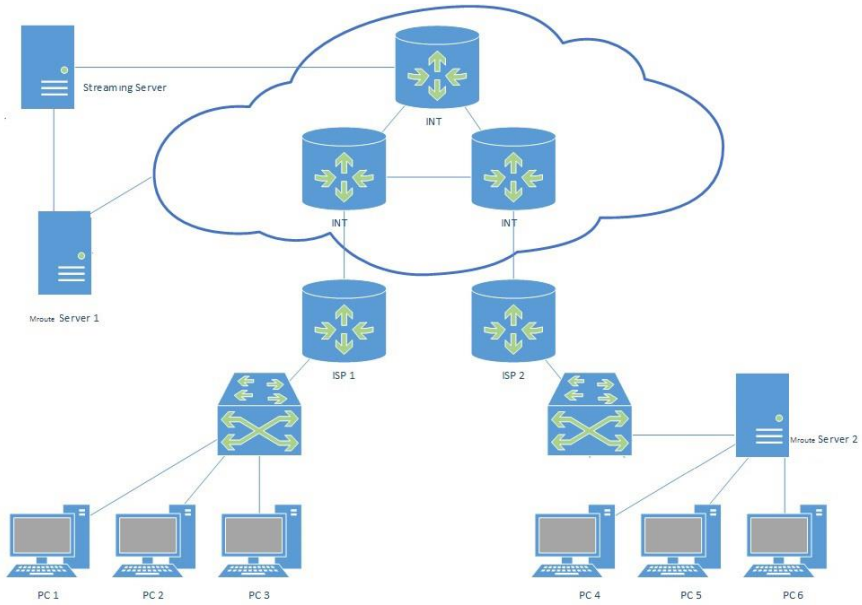


Figure 4.2 Multicasting using XORP Mroute for tunneling

In this network we will send unicast data through mroute server 1 to the specific mroute server 2. Mroute server 2 will handle the unicast data and convert into specific multicast data and distribute it into its network that can handle multicast.

4.2.2.1 XORP Setup

We need to install XROP mroute on the server that will handle the traffic, There are a few issues that must be taken care off before we run the router. First we need to have those interfaces that are specified in the configuration. This is necessary when establishing tunnels between multicast hosts separated by unicast-only networks and routers. (The mrouted is a daemon that implements the multicast routing algorithm - the routing policy- and instructs the kernel on how to route multicast datagrams) [13].

4.2.3 Multicast data through blocked using SSH tunnel

We will create architecture to receive multicast data from streaming server to other network using SSH tunnelling process. By making make a remote connection with the server present in the local network of the streaming server and forward request to it. When same way our “Server” as in figure below will receive the data and send it to the client via SSH tunnel [14].

For this we will use two features of SSH:

The SOCKS proxy function

The Remote port forwarding feature

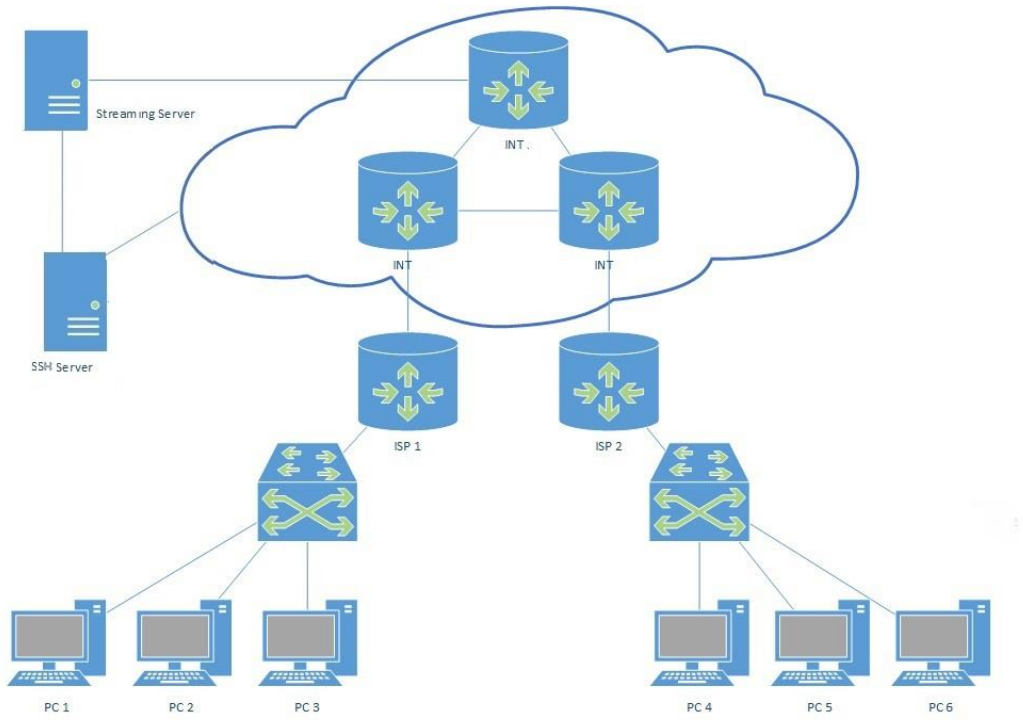


Figure 4.3 Multicasting Using SSH tunnel and port forwarding

4.2.3.1 SSH Server

SSH server will help us create a secure shell (SSH) tunnel consists of an encrypted tunnel created through a SSH protocol connection. Users may set up SSH tunnels to transfer unencrypted traffic over a network through an encrypted channel. For example, Microsoft Windows machines can share files using the Server Message Block (SMB) protocol, a non-encrypted protocol. If one were to mount a Microsoft Windows file-system remotely through the Internet, someone snooping on the connection could see transferred files. To mount the Windows file-system securely, one can establish a SSH tunnel that routes all SMB traffic to the remote fileserver through an encrypted channel. Even though the SMB protocol itself contains no encryption, the encrypted SSH channel through which it travels offers security [15].

To set up an SSH tunnel, one configures an SSH client to forward a specified local port to a port on the remote machine. Once the SSH tunnel has been established, the user can connect to the specified local port to access the network service. The local port need not have the same port number as the remote port.

SSH tunnels provide a means to bypass firewalls that prohibit certain Internet services so long as a site allows outgoing connections. For example, an organization may prohibit a user from accessing Internet web pages (port 80) directly without passing through the organization's proxy filter (which provides the organization with a means of monitoring and controlling what the user sees through the web). But users may not wish to have their web traffic monitored or blocked by the organization's proxy filter. If users can connect to an external SSH server, they can create a SSH tunnel to forward a given port on their local machine to port 80 on a remote web-server. To access the remote web-server, users would point their browser to the local port at http://localhost/.

Some SSH clients support dynamic port forwarding that allows the user to create a SOCKS 4/5 proxy. In this case users can configure their applications to use their local SOCKS proxy server. This gives more flexibility than creating a SSH tunnel to a single port as previously described. SOCKS can free the user from the limitations of connecting only to a predefined remote port and server. If an application doesn't support SOCKS, one can use a "socksifier" to redirect the application to the local SOCKS proxy server. Some "socksifiers" support SSH directly, thus avoiding the need for a SSH client.

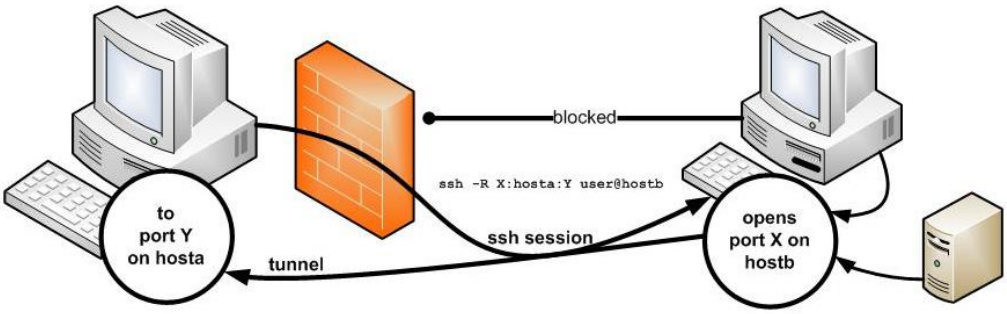


Figure 4.4 Reverse SSH Tunnel

4.2.3.2 SSH Application

Client side will need some tool to connect to SSH server, and application should be linked to tool to forward data to SSH server. Most common tool for SSH access in putty.

4.3 Compare Different Models discussed earlier

Both the models have different advantages and disadvantages. In context to architecture following metrics are considered:

1. **Independent:** In both the network architecture you are independent of the 3rd party solution to transmit multicast data.

2. **Easy to Setup:** Both the architecture are easy to setup and in SSH tunnel method you can make streaming server as a SSH server that will require no additional hardware. Mroute requires additional hardware to setup.

3. **Capacity**: It is the number of concurrent requests that can be handled by the server a given time period are different for both. For mroute method it can be increased by increasing power of server. But in case of SSH server tunnel it depends upon number of ports available as it has limited number of ports. It is not use full if we have many different network and hosts.

4.6 Conclusion

In this chapter we studied two network architecture solves issues of sending multicast data over internet without the use of 3rd party solution. With the help of two different ways of tunnelling we can setup the server. According to the requirements we can implement the architecture. Next section we implements the proposed architecture.

Chapter 5

Implementation

This chapter focuses on implementation of proposed architecture. First experimental setup is explained covering all hardware requirements for implementation. Next tools used for implementation are explained briefly.

5.1 Experimental Setup

Experiments are conducted to setup the proposed architecture on machines, network simulation using GNS3 and public VPS was used.

5.2 Tools Used

This section gives some brief of tools used to implement proposed architecture.

5.2.1 Virtual Box

VirtualBox is a powerful x86 and AMD64/Intel64 virtualization product for enterprise as well as home use. Not only is VirtualBox an extremely feature rich, high performance product for enterprise customers, it is also the only professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 2 [16].

Presently, VirtualBox runs on Windows, Linux, Macintosh, and Solaris hosts and supports a large number of guest operating systems including but not limited to Windows (NT 4.0, 2000, XP, Server 2003, Vista, Windows 7), DOS/Windows 3.x, Linux (2.4 and 2.6), Solaris and OpenSolaris, OS/2, and OpenBSD.

VirtualBox is being actively developed with frequent releases and has an ever growing list of features, supported guest operating systems and platforms it runs on. VirtualBox is a community effort backed by a dedicated company: everyone is encouraged to contribute while Oracle ensures the product always meets professional quality criteria.

5.2.2 XORP

XORP is an open source Internet Protocol routing software suite originally designed at the International Computer Science Institute in Berkeley, California. The name is derived from extensible open router platform.

The product is designed from principles of software modularity and extensibility and aims at exhibiting stability and providing feature requirements for production use while also supporting networking research. The development project was founded by Mark Handley in 2000. Receiving funding from Intel, Microsoft, and the National Science Foundation, it released its first production software in July 2004. The project was then run by Atanu Ghosh of the International Computer Science Institute, in Berkeley, California [13].

In July 2008, the International Computer Science Institute transferred the XORP technology to a new entity, XORP Inc., a commercial startup founded by the leaders of the opensource project team and backed by Onset Ventures and Highland Capital Partners. In February 2010, XORP Inc. was wound up, a victim of the recession. However the open source project continued, with the servers based at University College London. In March 2011, Ben Greear became the project maintainer and the www.xorp.org server is now hosted by Candela Technologies.

The XORP codebase consists of around 670,000 lines of C++ and is developed primarily on Linux, but supported on FreeBSD, OpenBSD, DragonFlyBSD, NetBSD. Support for XORP on Microsoft Windows was recently re-added to the development tree. XORP is available for download as a Live CD or as source code via the project's homepage.

The software suite was selected commercially as the routing platform for the Vyatta line of products in its early releases, but later has been replaced with quagga.

XORP provides a command line interface for interactive configuration and operation monitoring. The interface is implemented as a distinct application called xorpsh, that may be invoked by multiple users simultaneously. It interacts via Interprocess communication with the router core modules. The command line language is modelled after that of Juniper Networks's JunOS platform [17].

5.2.3 µTorrent

The program was designed to use minimal computer resources while offering functionality comparable to larger Torrent clients such as Vuze or BitComet. The program has received consistently good reviews for its feature set, performance, stability, and support for older hardware and versions of Windows.

The program has been in active development since its first release in 2005. Although originally developed by Ludvig Strigeus, since December 7, 2006, the code is owned and maintained by BitTorrent, Inc. The code has also been employed by BitTorrent, Inc. as the basis for version 6.0 and above of the BitTorrent client, a re-branded version of µTorrent [18].

5.2.4 Putty

PuTTY is a free and open source terminal emulator application which can act as a client for the SSH, Telnet, rlogin, and raw TCP computing protocols and as a serial console client. PuTTY was originally written for Microsoft Windows, but it has been ported to various other operating systems. Official ports are available for some Unix-like platforms, with work-in-progress ports to Classic Mac OS and Mac OS X, and unofficial ports have been contributed to platforms such as Symbian and Windows Mobile [19].

Some features of Putty are:

- The storing of hosts and preferences for later use.
- Control over the SSH encryption key and protocol version.
- Command-line SCP and SFTP clients, called "pscp" and "psftp" respectively.
- Control over port forwarding with SSH (local, remote or dynamic port forwarding), including built-in handling of X11 forwarding.
- Emulates most xterm, VT102 control sequences, as well as much of ECMA-48 terminal emulation.
- Supports 3DES, AES, Arcfour, Blowfish, DES.
- Public-key authentication support (no certificate support).
- Support for local serial port connections.
- Self-contained executable requires no installation.

5.3 Implementation

This section shows how to implement the proposed architecture using tools explain in above section.

5.3.1 XORP setup on Ubuntu

We will use Multicasting Linux MRouter with XORP we will follow. Steps to install XORP and configure it [20].

Step 1: Check Linux Kernel Functions

CONFIG_IP_MULTICAST=y

CONFIG_IP_ADVANCED_ROUTER=y

CONFIG_IP_MROUTE=y

After this configurations, the linux kernel support IGMP, DVMRP and MOSPF. If want support PIM-SM , do below settings

CONFIG_IP_PIMSM_V2=y

PS: you can use vi view /boot/CONFIG-(kernel version) to confirm

Step 2: edit /etc/sysctl.conf to enable ip_forward

net.ipv4.conf.default.forwarding=1

PS: reboot or use command “sysctl -p” to enable !

Step 3: Install XORP(To compile XORP requires nearly 1.4GB of free disk space)

Step 3.1: Download xorp-1.6.tar.gz

Step 3.2: untar the xorp-1.6.tar.gz

Step 3.3: ./configure

Step 3.4: make

Step 4: run rtrmgr/xorp_rtrmgr rtrmgr/xorpsh (A sample xorp command shell)

Step 4.1: create user group named “xorp”

sudo addgroup xorp

Step 4.2: cp or edit config.boot in xorp-1.6/rtrmgr/config.boot config.boot for our environment

protocols {

fib2mrib {

disable: false

}

igmp {

disable: false

interface eth0 {

vif eth0 {

disable: false

version: 3

enable-ip-router-alert-option-check: false

query-interval: 125

query-last-member-interval: 1

query-response-interval: 10

robust-count: 2

}}}}

fea {

unicast-forwarding4 {

disable: false

}}

interfaces {

restore-original-config-on-shutdown: false

interface eth0 {

disable: false

discard: false

description: " TEXT"

default-system-config {

}}}

Step 4.3: xorp-1.6/rtrmgr/xorp_rtrmgr -b xorp-1.6/rtrmgr/config.boot

Step 5: Check Mrouter work OK

Step 5.1: Connect eth0 to VLC Player server PC and connect eth1 to VLC Player client PC

Step 5.2: Use VLC Player server PC run multicast streaming

Step 5.3: Use VLC Player client PC to play multicast streaming by multicast URL

Step 6: run rtrmgr/xorpsh (A sample xorp command shell)

The steps will help configure XORP for multicast routing.

5.3.2 Media Server Windows Server 2008

We will show you how to setup windows server 2008 as media stream server with ability to handle multicast request. First the choice of which server type to install. Since Windows Media services require at least Enterprise or Datacenter to provide multicast it is important to choose at least one of those. Then we select type of service required we select stream server and webserver IIS. Next option is to play movie on demand or simply broadcast and play user at its current broadcast point as in television [21].

After that we select the server option to handle request as a multicast server, then we select files to record the multicast group information. In the last part we have the file path which we want to multicast that will handle request of the incoming connection.

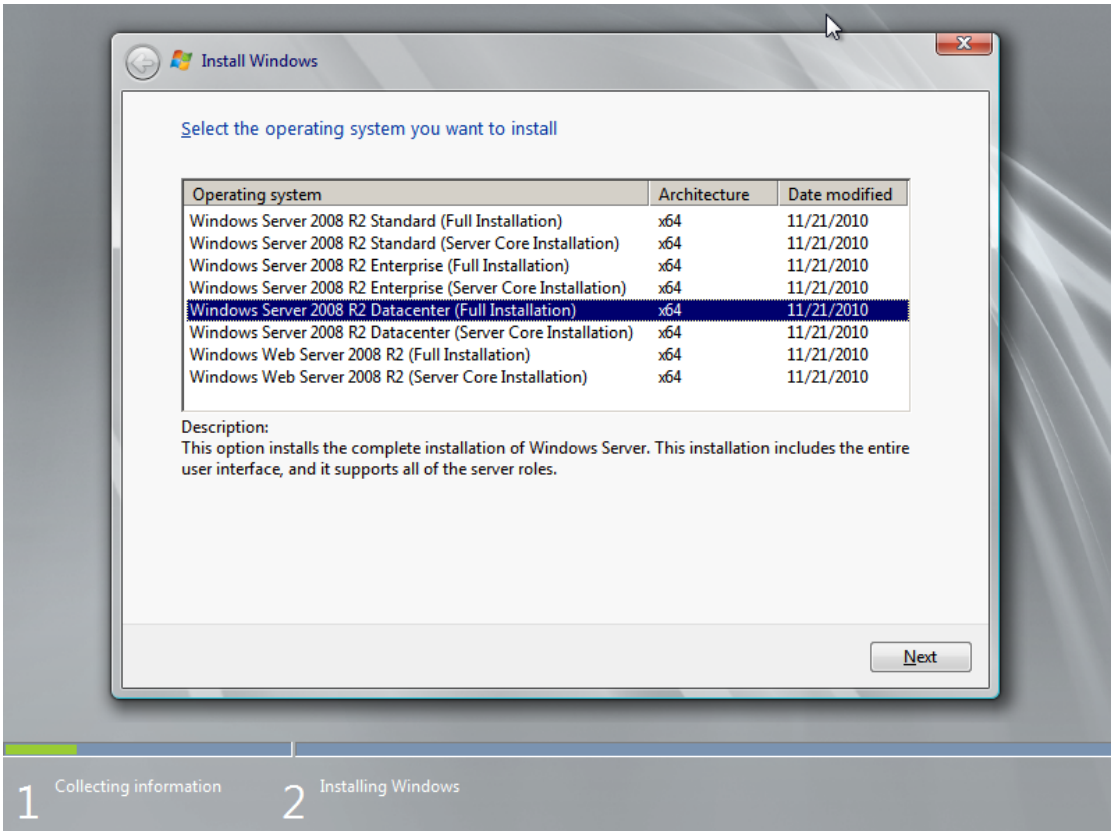


Figure 5.1 Choose which type of server to install

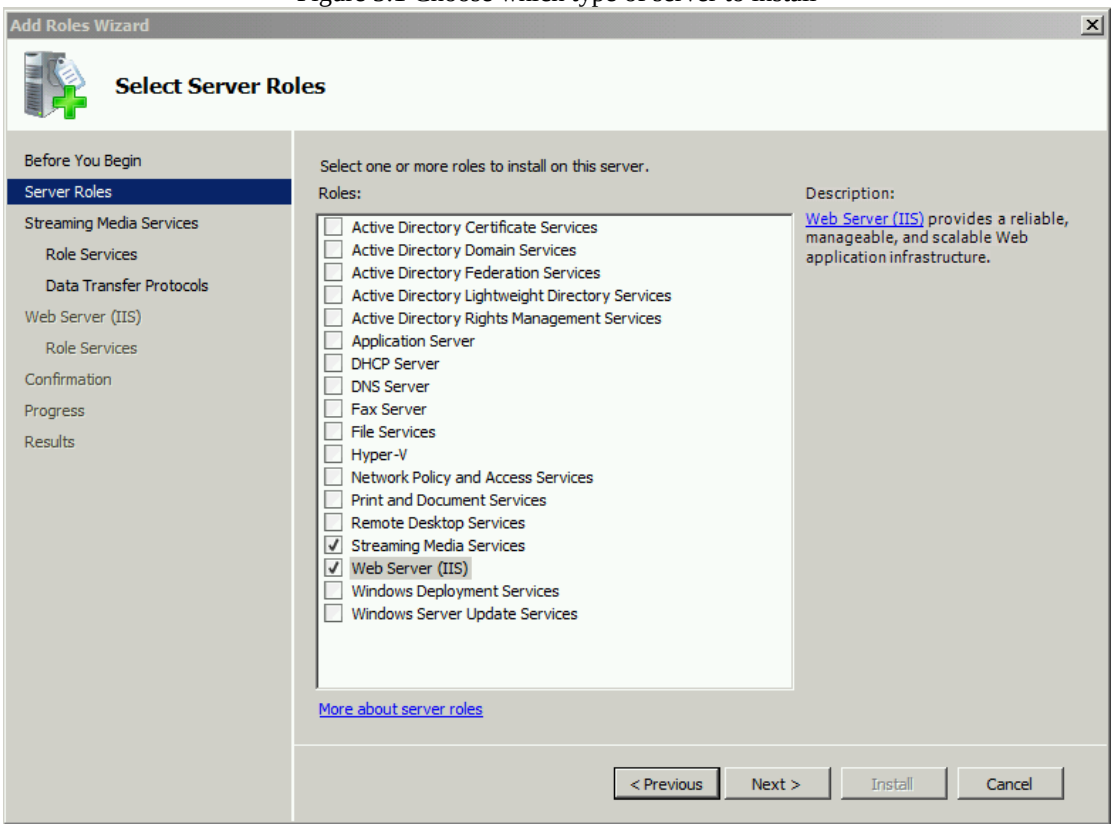


Figure 5.2 Select server services

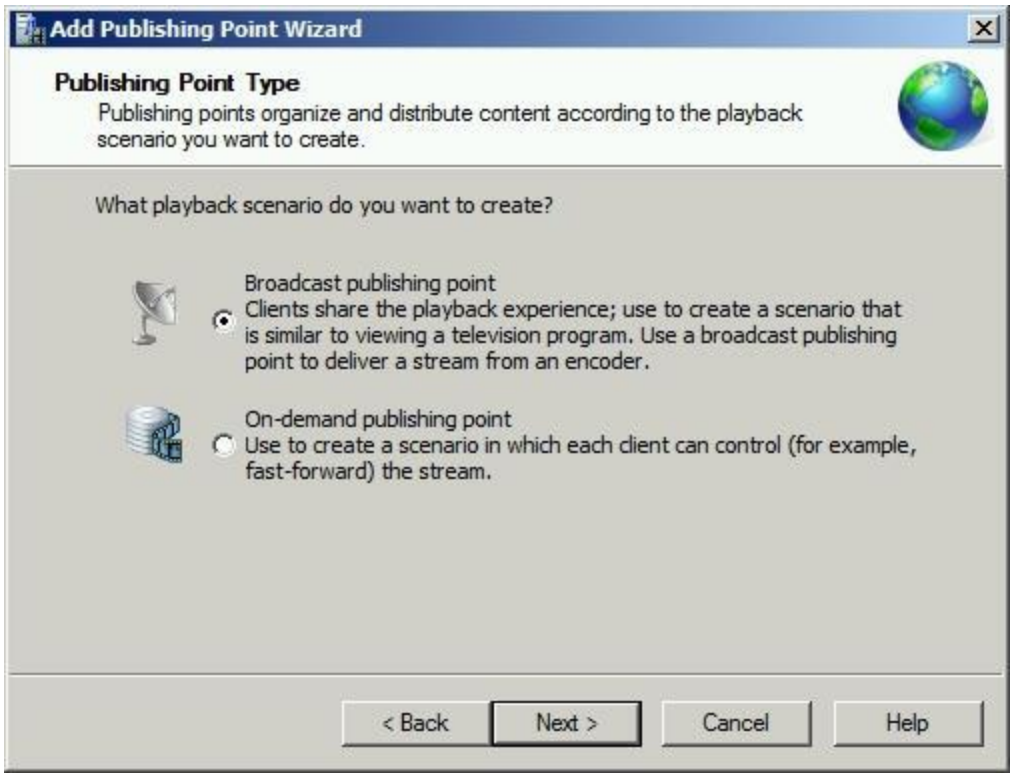


Figure 5.3 Choose playback scenario

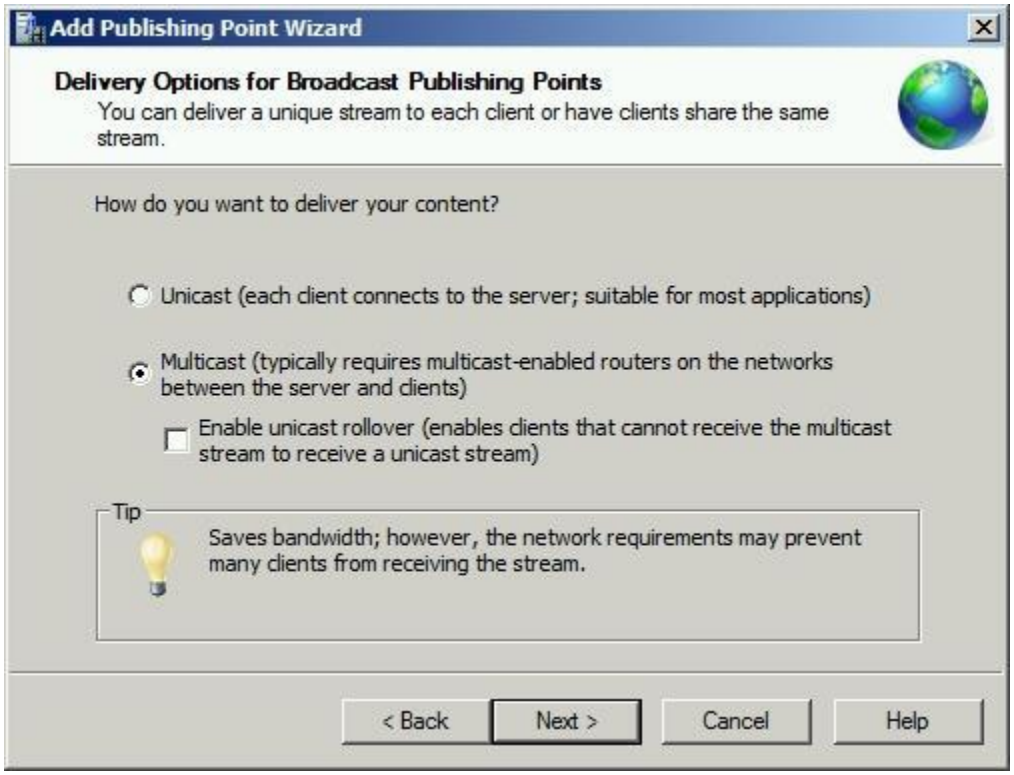


Figure 5.4 How to deliver content

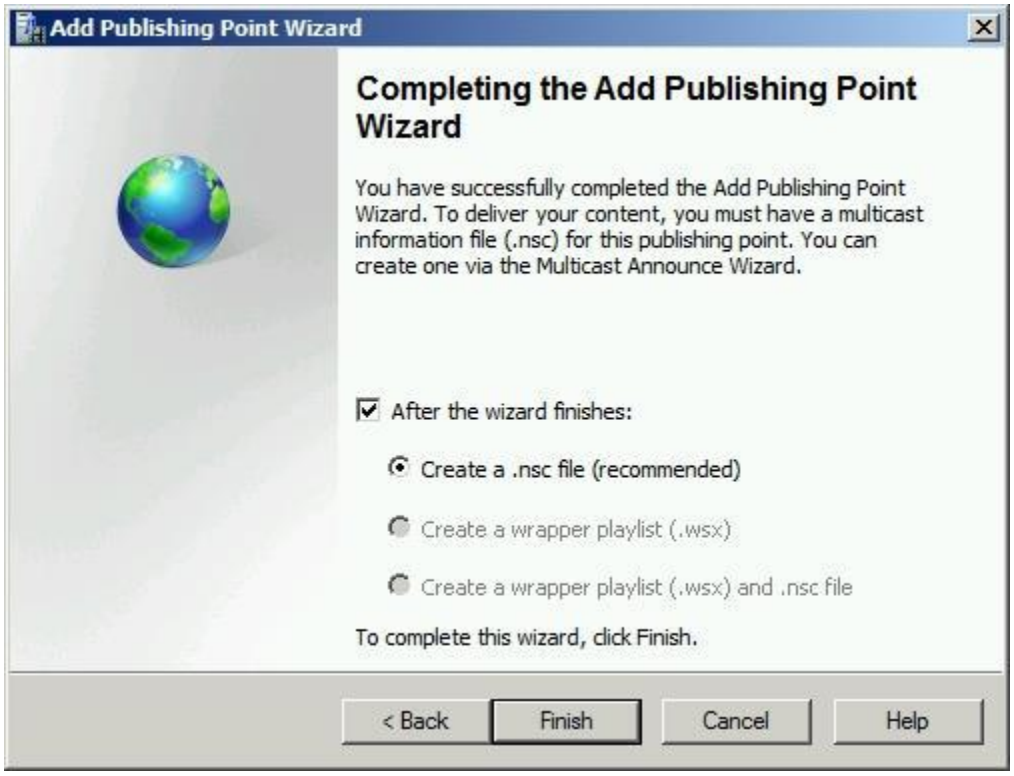


Figure 5.5 Add publishing point

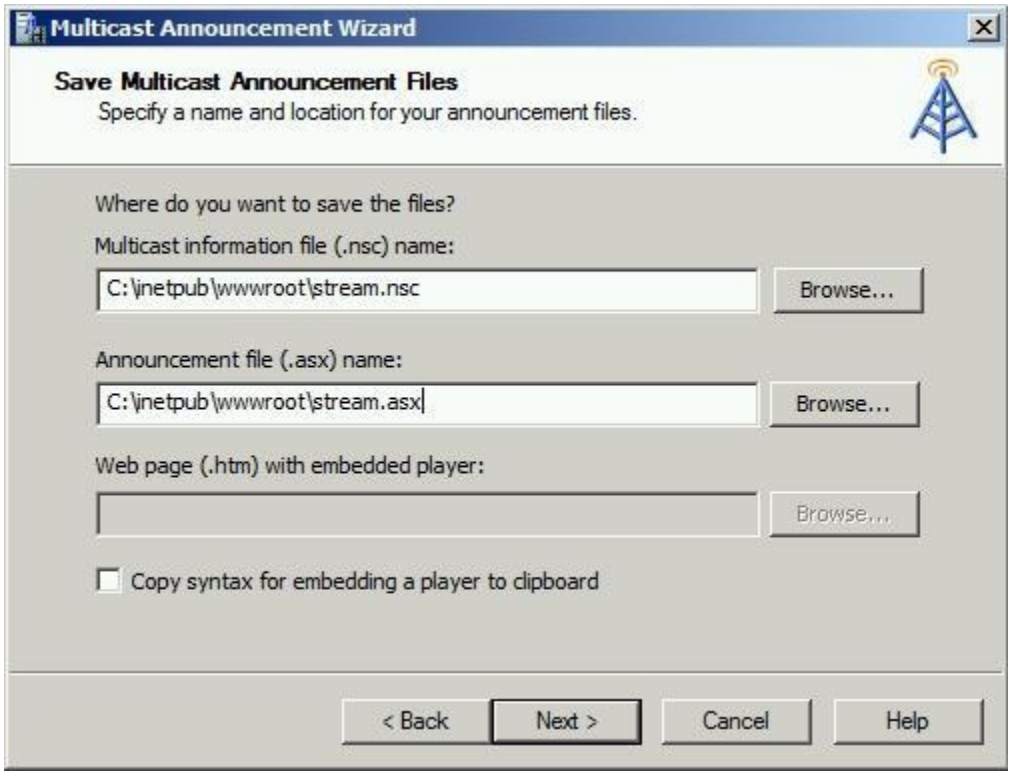


Figure 5.6 Add announcement file

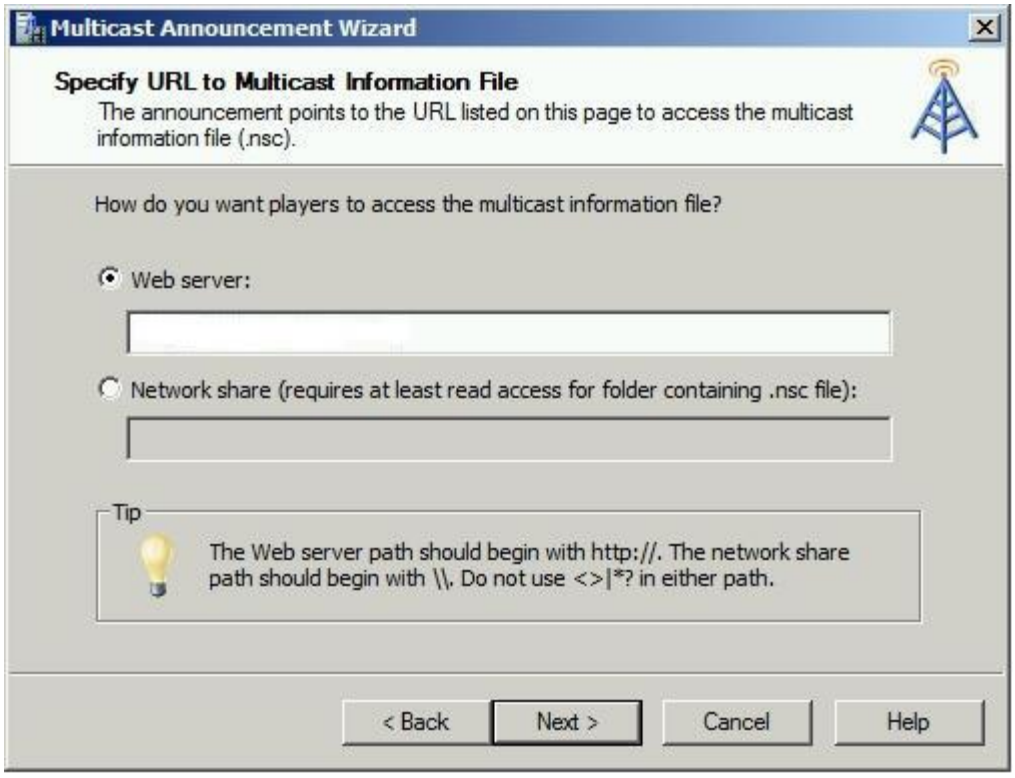


Figure 5.7 Select Media file to stream

5.3.3 uTorrent Connect to SSH server

To show the SSH tunnel we will configure torrent client as there no open source application for getting multicast data from desired server to connect to desired server using SSH. In this we will show how it connect to SSH server. You need an SSH account on the server in order to get this working. In figure 5.8 we can see the the putty console to connect to desired server. After that you can open your SSH server and on the server we must configure the ssh daemon to accept and forward incoming connections from other hosts than localhosts [22]. This is done by setting the GatewayPorts configuration directive to "yes". The server in running, so I included the line:

GatewayPorts yes

In the file /etc/ssh/sshd_config, which can be different for different linux version so we can use locate command to find the this file. After a configuration change has taken place the sshd will need to be restarted. The can be done by running "killall –HUP sshd" as root.

After has been done we can now setup the tunnel on the client by running the command in a shell or "Terminal" window as root:

ssh -2 -R 7654:localhost:7654 -A -D 1080 root@<SERVER IP or HOSTNAME>

This command is bind command it will open port 1080 on the client and tunnel traffic from there to a SOCKS proxy server on the server and Setup a server socket on the server listening on port 7654 and forward any connection from here back to the same port on the client. This command will help to bind the utorrent data on local machine to putty and then putty which is connected to server will get the required data from that network.

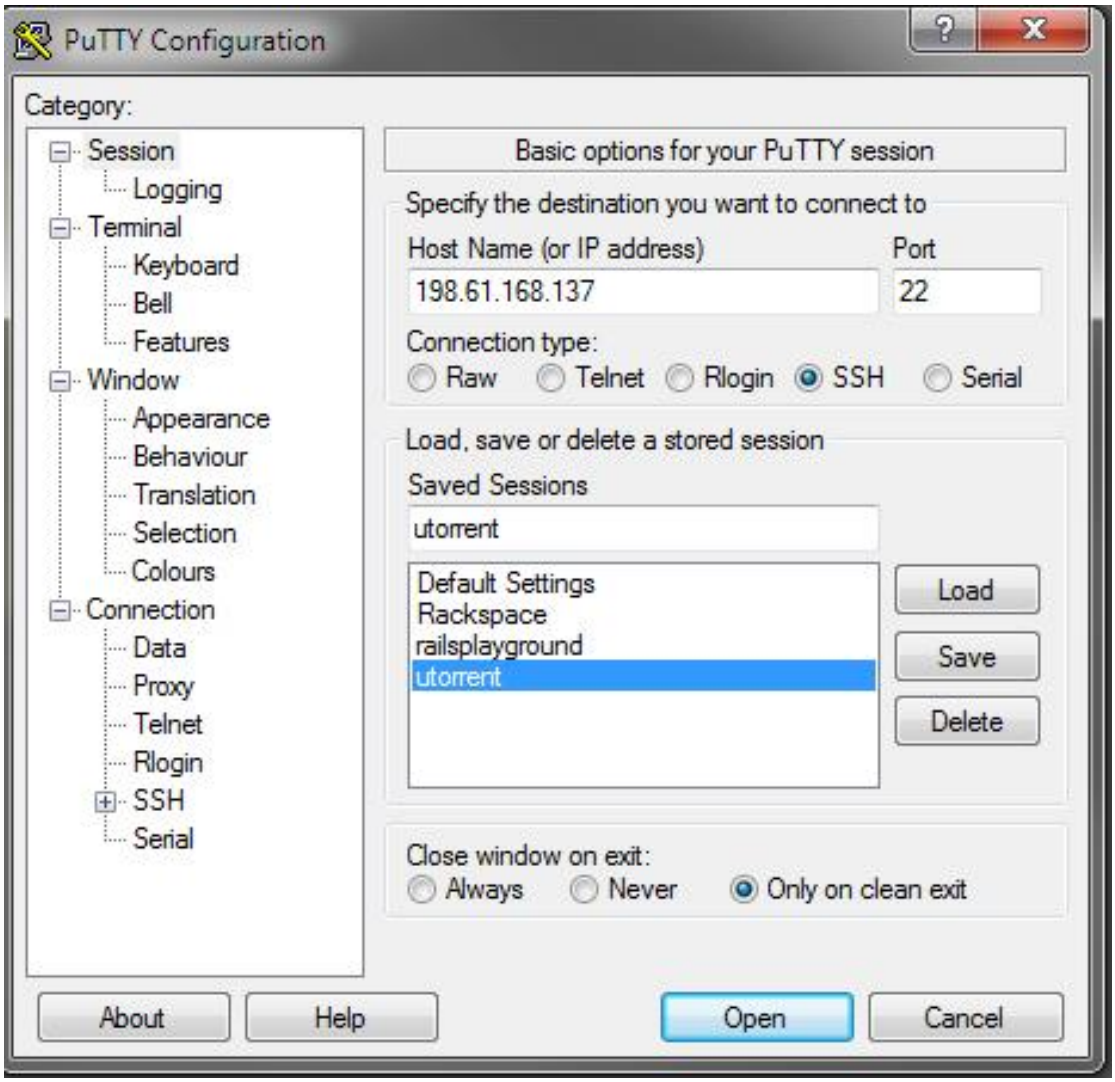


Figure 5.8 Putty terminal

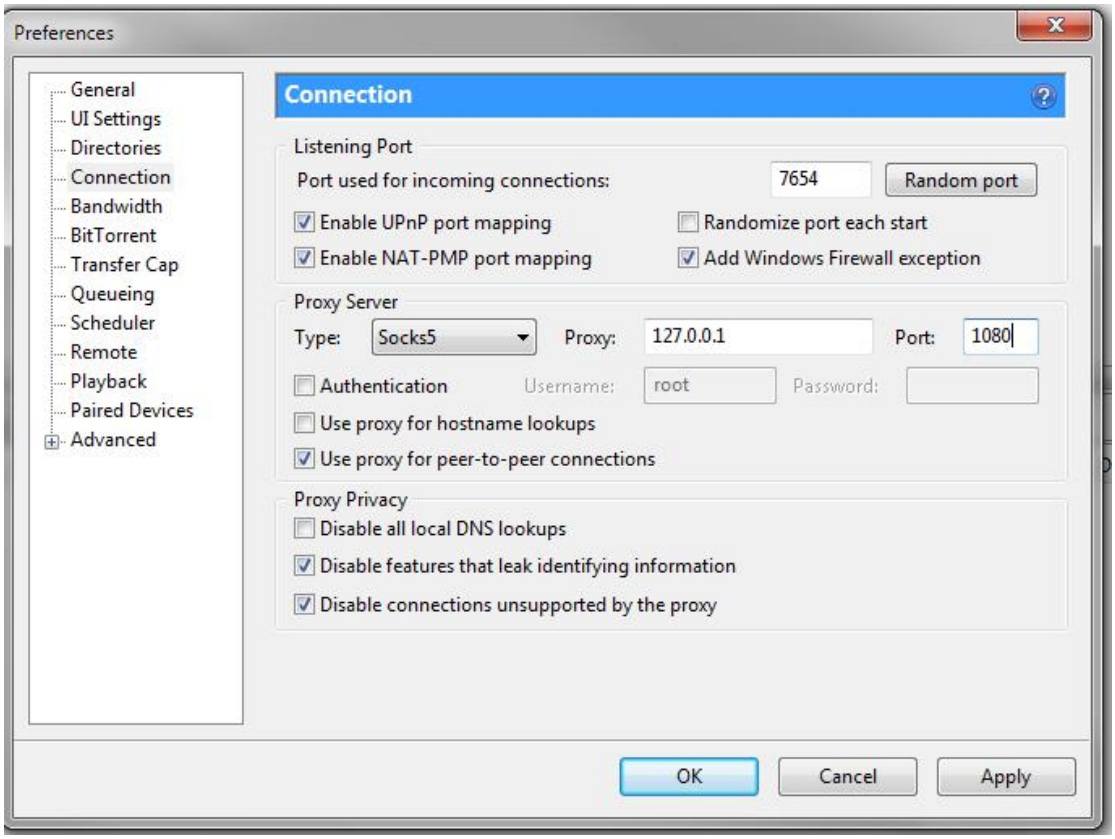


Figure 5.9 utorrent setting to forward data to putty

Similarly, we can develop application can connect to server that can be streaming server itself or any local server in that network to receive the multicast data.

5.3.4 Conclusion

This section demonstrates the how we can configure server to multicast data over internet. Tools used for demonstration are explained briefly then how can we setup the architecture's explained in earlier chapter.

Chapter 6
Conclusion

This Chapter concludes the work presented in this thesis. At the end of this chapter, some future directions have been proposed which can be considered to improve the Multicast architecture.

6.1 Conclusion

This thesis provides a new method of multicasting data by creating own tunnelling m method without using any other service. Which is essential for the organisation as there is no set solution for tunnelling data over internet Given that benefits of the multicasting over unicast in bandwidth utilization. As future of broadcasting is multicast in IPv6 more people organisation will dig into finding solution for it. This presented architecture includes four main improvements:

1. Setup secures tunnelling process.
2. Easy to setup and maintain with minimum hardware requirement.
3. Reliable as you have control of the network.
4. Configurable as per requirement.

One of the desirable attribute of our architecture is it give users freedom to send data to using multicast protocol.

6.2 Thesis Contribution

Some of thesis contributions are:

I. Literature review is done in two different domains of Multicasting viz. Multicast data over internet and setting own tunnelling system..
II. Developed using existing tools expect for streaming application to connect to putty or ssh server that also for video streaming.
III. A method is developed for better resource discovery and matching process.
IV. Designed architecture can has can be used by different OS systems to implement.

6.3 Future Scope

There are some aspects of architecture that need to be discussed:

1. If organisation wants to commercialize this architecture then they need different mechanism to handle IGMP message.
2. Different types of files can be streamed using this method.
3. Development of application architecture using SSH tunnel to connect.

Bibliography

[1] Gorry Fairhurst. (2009, March) The University of Aberdeen. [Online]. http://www.erg.abdn.ac.uk/~gorry/eg3567/intro-pages/uni-b-mcast.html

[2] Beau Williamson, *IP Multicast Networks*., 2003.

[3] Abley J. (2013, March) Operation of Anycast Services. [Online]. http://tools.ietf.org/pdf/rfc4786.pdf

[4] T. Imielinski. (1996) Network Working Group. [Online]. http://tools.ietf.org/html/draft-rfced-exp-navas-00

[5] L. Mohamed. (1997) Carleton University Instructional Television Courses. [Online]. http://www.sce.carleton.ca/tln/ITVMBONE.htm

[6] Mike Macedonia. MBONE, the Multicast BackbONE. [Online]. http://www-mice.cs.ucl.ac.uk/multimedia/projects/mice/mbone_review.html

[7] Fast Lane. [Online]. http://www.fastlaneus.com/course/ci-mcast

[8] Bob Fink. IETF. [Online]. http://www.ietf.org/wg/concluded/6bone.html

[9] Juniper. [Online]. http://www.octoshape.com/wp-content/uploads/2013/06/Juniper-Octoshape-White-Paper.pdf

[10] IBM. [Online]. http://publib.boulder.ibm.com/infocenter/iseries/v5r3

[11] Chuck Semeria, "Introduction to IP Multicast Routing," *NC State University*.

[12] Juniper. [Online]. http://www.juniper.net/techpubs/software/junos-es/

[13] Xorp. [Online]. http://www.xorp.org/getting_started.html

[14] Revolution Systems. [Online]. http://www.revsys.com/writings/quicktips/ssh-tunnel.html

[15] OpenSSH. [Online]. http://www.openssh.org/

[16] Virtual Box. [Online]. https://www.virtualbox.org/

[17] Git Hub. [Online]. https://github.com/greearb/xorp.ct

[18] utorrent. [Online]. http://www.utorrent.com/get-started

[19] Putty. [Online]. http://www.putty.org/

[20] Ubuntu Manuals. [Online]. http://manpages.ubuntu.com/manpages/hardy/man8/xorp_rtrmgr.8.html

[21] Microsoft Support. [Online]. http://support.microsoft.com/kb/963697

[22] Media Template. [Online]. https://kb.mediatemple.net/questions/16/

LIST OF PUBLICATIONS

[1] Amandeep Singh & Dr. Neeraj Kumar, "An Architecture for Multicasting using Private Tunnel ", *International Journal of Engineering Research and Technology,* [Communicated]

[2] Amandeep Singh & Dr. Neeraj Kumar, "Open source Website Security and encrypted hack detection", *International Journal of Engineering Research and Technology,* [Communicated]