MEASUREMENT BASED COMPARISON BETWEEN VoIPoMPLS AND VoIP USING SOFTWARE ROUTERS

by

Itrat Rasool Quadri

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

In Partial Fulfillment of the Requirements for the
Degree

MASTER OF SCIENCE
in

Computer Networks

King Fahd University of Petroleum & Minerals

Dhahran, Saudi Arabia

October 2004

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **ITRAT RASOOL QUADRI** under the direction of his thesis advisor and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER NETWORKS**.

Thesis Committee

_____
Dr. AbdulWaheed M. A. Sattar (Chairman)

_____
Dr. Abdulaziz S. Al-Mulhem (Member)

_____
Dr. Mohammed H. Sqalli (Member)

_____
Department Chairman
Prof. Sadiq M. Sait

_____
Dean of Graduate Studies
Dr. Mohammad A. Al-Ohali

_____
Date

DEDICATION

This thesis is dedicated to my parents and my brother

Nazahat Rasool Quadri

Ashraf Quadri

Midhat Rasool Quadri

ACKNOWLEDGMENTS

Acknowledgement is due to King Fahd University of Petroleum & Minerals for supporting this research. My sincere appreciation goes to Dr. AbdulWaheed M. A. S, for his efforts, advice, encouragement and invaluable support given as my advisor. I also wish to thank my thesis committee members Dr. Abdulaziz S. Al-Mulhem and Dr. Mohammed H. Sqalli for their help, support, and contributions. In addition, I appreciate the support rendered to me by the department Chairman, Dr. Sadiq Sait.

I would like to thank James R. Leu for various discussions and comments during the early stage of my work. In addition, my fellow Performance Engineering Laboratory (PEL) associates, Amisu and Rehan for the great time we shared in the lab.

I acknowledge my colleagues and the loving support and encouragement from my parents and brother without whom I certainly would not have made it this far.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

THESIS ABSTRACT (ENGLISH)

NAME: Itrat Rasool Quadri

TITLE: MEASUREMENT BASED COMPARISON BETWEEN VoIPoMPLS AND VoIP USING SOFTWARE ROUTERS

MAJOR FIELD: Computer Networks.

DATE OF DEGREE: October 2004.

Growing demand for Voice over Internet Protocol (VoIP) services has motivated the solution providers to look for technologies that provide better, more manageable and efficient delivery of voice traffic. Multi-Protocol Label Switching (MPLS) is one such technology. Voice over Internet Protocol over Multi-Protocol Label Switching (VoIPoMPLS) is a relatively new idea in facilitating voice traffic between access networks over an MPLS backbone as opposed to VoIP that uses the best-effort Internet for transporting voice traffic. An MPLS network provides comprehensive options for Quality of Service (QoS) and signaling of voice traffic compared to Frame Relay and IP networks. IP is a major competitor of MPLS in the access networks as voice may already be in IP packets when it reaches the Integrated Access Device. In this thesis we show the benefit provided by an MPLS Network over an IP Network in terms of delay, jitter, throughput and other performance metrics like CPU utilization, virtual memory and interrupts/sec for voice traffic.

THESIS ABSTRACT (ARABIC)

الأسم: عترت رسول قادري
العنوان: مقارنة باستخدام القياسات بين تقنيتي نقل الصوت بواسطة تبديل الواسمة المتعدد البروتوكولات (VoIPoMPLS) و نقل الصوت بواسطة بروتوكول الإنترنت (VoIP) باستخدام المحولات البرمجية.
الحقل: شبكات الحاسب الآلي
تاريخ نيل الدرجة: أكتوبر 2004

إن الإحتياج المتزايد لخدمة نقل الصوت بواسطة بروتوكول الإنترنت أو ما يعرف بالـ Voice over IP (VoIP) شجع موفري الحلول على البحث عن تقنيات تمكن من الحصول على خدمات ذات جودة و كفاءة عاليتيين. أحد هذه التقنيات تعرف بـتبديل الواسمة المتعدد البروتوكولات أو Multi Protocol Label Switching (MPLS). إن فكرة نقل الصوت بواسطة تقنية ال MPLS المعروفة ب (VoIPoMPLS) هي فكرة جديدة تهدف إلى الإستفادة من المميزات التي توفرها التقنية الآنفة الذكر من نقل الصوت عبر الشبكات الأساسية مقارنة بالخدمة التي يوفرها بروتوكول (VoIP) التي تعتمد مفهوم (أفضل جهد). إن الشبكات التي تشغّل بإستخدام تقنية الـ (MPLS) توفر قائمة طويلة من الخيارات لكفاءة الخدمة و نقل الصوت مقارنة بـالشبكات التي تستخدم تقنيتي نقل الإطار المرحلي (Frame Relay) و بروتوكول الإنترنت (IP). إن تقنية بروتوكول الإنترنت تمثل منافساً قوياً لتقنية الـ MPLS نظراً لأن أجهزة الإستقبال المتكاملة في الشبكات تستقبل حزم الصوت و هي معالجة ببروتوكول الإنترنت. في هذه الرسالة سنوضح الميزات التي توفرها تقنية الـ MPLS مقارنة بتقنية بروتكول الإنترنت من حيث التأخير, التأخير النسبي, معدل التشغيل بالإضافة إلى معايير أخرى كـالإستفادة من المعالج و الذاكرة الوهمية و عدد المقاطعات في الثانية التي يتعرض لها لصوت المنقول.

درجة الماجستير في العلوم
جامعـة الملك فهـد للبترول والمعـادن ، الظهران – السعودية
أكتوبر 2004

CHAPTER 1

1 INTRODUCTION

Three complimentary technical instruments have been employed by service providers to accommodate Internet growth [1], which are:

- Network architecture
- Capacity Expansion
- Traffic Engineering

Network architecture encompasses abstract structure of networks: components of the network, their functions and their interrelationship. Large Internet Service Providers (ISPs) have responded to traffic growth through expansion of network capacity and infrastructure. Finally, traffic engineering has been employed to address Internet traffic growth. With time it has been learned that simple capacity expansions are essential but not sufficient to deliver high quality service under varying traffic conditions. Traffic engineering addresses performance optimization of operational networks. It applies scientific principles of measurement, modeling, characterization, and control of Internet traffic to achieve reliable and efficient movement of traffic through the network, optimal resource utilization and effective network capacity planning.

Limited functional capabilities of conventional IP technologies have made it difficult for the effective implementation of traffic engineering in public IP networks. Measurement functions such as traffic matrix required for traffic engineering are difficult to estimate from interface statistics on IP routers. Interior Gateway Protocols (IGPs), such as Intermediate System-to-Intermediate System (IS-IS) and Open Shortest Path first (OSPF) used for routing traffic within autonomous systems are topology driven and require progressive connection control. Routers make independent routing decisions based on link state database. Route selection depends on shortest path computations. The approach is scalable but not without drawbacks. The problem is that these protocols do not take into consideration the characteristics of offered traffic and network capacity constraints when making routing decisions. The result is understandably the underutilization of several resources. This is the specific area, which traffic engineering addresses.

Multi Protocol Label Switching (MPLS) supports traffic engineering. In MPLS a virtual connection is established between two points on a datagram network. The MPLS connection is the Label Switched Path (LSP). LSPs are used in a manner similar to a connection-oriented network while still retaining the efficiency of the underlying network. MPLS can emulate connection oriented network as well as hybrid architecture where LSPs are used to deliver connection-oriented services but datagram mechanisms are used to deliver datagram services. Hybrid architecture excels performance-wise as there is less management overhead with connection emulation, which minimizes the cost of running the network [1].

1.1 Software Routers

Software routers are slower and more unreliable than their hardware counterparts but they offer connectivity at a far lower price. Bridging a workgroup onto an existing LAN can be done using an old Pentium and adding to it an ethernet card and some free software. Internet connectivity can be achieved without the purchase of an Integrated Services Digital Network (ISDN) router, by employing a simple modem to the software router.

What is important from a router's point of view is the number of packets that can be routed in a given time slice, which is related to the router's backplane capacity. In large high bandwidth environment hardware based routers are the obvious choice because of their capability to handle large loads. However if a situation demands the use of routers and the money to finance their purchase is not an option then a PC with several ethernet cards can be used to set up a software router. Software routers can be deployed in a network with a tight budget. In case of software routers a single point of entry to the network allows the implementation of a firewall to protect the entire ethernet from exterior intruders.

1.2 IPv4 and MPLS

The Internet Protocol is the key tool used today to build scalable, heterogeneous Internetworks. It runs on all the nodes (both hosts and routers) in a collection of networks and defines the infrastructure that allows these nodes and networks to function as a single logical Internetwork. The IP service model has two parts; an addressing scheme, which

provides a way to identify all hosts in the network, and a datagram (connectionless) model of data delivery.

The IP datagram is sent in a connectionless manner over a network. Every datagram carries enough information to let the network forward the packet to its correct destination; there is no need for any advance setup mechanism to tell the network what to do when the packet arrives. You just send it, and the network makes its best effort to get it to the desired destination.

Keeping the routers as simple as possible was one of the original design goals of IP. The ability of IP to "run over anything" is often cited as one of its most important characteristics.

Best effort delivery does not just mean that packets can get lost. Sometimes packets do get delivered out of order, and sometimes the same packet gets delivered more than once. The higher-level protocols or applications that run above IP need to be aware of all these possible failure modes. The fact is that IP gives no guarantees. The IP datagram, like most packets, consists of a header followed by a number of bytes of data called payload. There is a need for a global addressing scheme to enable identification of all the hosts. Global uniqueness is the first property that should be provided in an addressing scheme.

IP addresses are hierarchical, which means that they are made up of several parts that correspond to some sort of hierarchy in the Internetwork. Specifically, IP addresses consist of two parts, a network part and a host part. The network part of an IP address identifies the network to which the host is attached; all hosts attached to the same network

have the same network part in their IP address. The host part then identifies each host uniquely on that particular network.

Forwarding is the process of taking a packet from an input and sending it out on the appropriate output, while routing is the process of building up the tables that allow the correct output for a packet to be determined. There are some important points to consider during the forwarding of IP datagrams:

- Every IP datagram contains the IP address of the destination host.
- The "network part" of an IP address uniquely identifies a single physical network that is part of the larger Internet.
- All hosts and routers that share the same network part of their address are connected to the same physical network and can thus communicate with each other by sending frames over that network.
- Every physical network that is part of the Internet has at least one router that, by definition, is also connected to at least one other physical network; this router can exchange packets with hosts or routers on either network.
- Forwarding IP datagrams can therefore be handled in the following way. A datagram is sent from a source host to a destination host, possibly passing through several routers along the way. Any node, whether it is a host or a router, first tries to establish whether it is connected to the same physical network as the destination. To do this, it compares the network part of the destination address with the network part of the address of each of its network interfaces. (Hosts normally have only one interface, while routers normally have two or more, since

they are typically connected to two or more networks.) If a match occurs, then that means that the destination lies in the same physical network as the interface, and the packet can be directly delivered over that network.

- If the node is not connected to the same physical network as the destination node, then it needs to send the datagram to a router. In general, each node will have a choice of several routers, and it needs to pick the best one, or at least one that has a reasonable chance of getting the datagram closer to its destination. The router that it chooses is known as the next hop router. The router finds the correct next hop by consulting its forwarding table. The forwarding table is conceptually just a list of <NetworkNum, NextHop> pairs. (In practice, forwarding tables often contain some additional information related to the next hop.) Normally, there is also a default router that is used if none of the entries in the table match the destination's network number. For a host, it may be quite acceptable to have a default router and nothing else – this means that all datagrams destined for hosts not on the physical network to which the sending host is attached will be sent out through the default router.
- To achieve scalability, we need to reduce the amount of information that is stored in each node and that is exchanged between nodes. The most common way to do that is hierarchical aggregation. IP introduces a two-level hierarchy, with networks at the top level and nodes at the bottom level. Aggregated information is obtained by letting routers deal only with reaching the right network; the information that a router needs to deliver a datagram to any node on a given network is represented by a single aggregated piece of information.

MPLS is a label swapping and forwarding technology. Packet headers or label values are changed as the packet moves from one node to another. It allows resources to be reserved and routes pre-determined. It provides connection-oriented functionality over connectionless IP network. Nodes at the edge of the network are connected through virtual links or tunnels. The basic idea is not to restrict MPLS to any link layer technology like Asynchronous Transfer Mode (ATM) or Frame Relay (FR). It is a standard for speeding up network traffic flow and provides ease of management. It involves setting up of a path for a given sequence of packets, identified by labels. This saves up time for a router to look up the address to the next node to forward the packet to.

MPLS will help in adding a number of capabilities to today's best effort IP networks which includes

- Layer 2 (Ethernet, ATM, FR) Virtual Private Network (VPN)s.
- Optical control plane for optical transport networks and solution of problems faced by networks
- Fast data link layer restoration.
- Integration of data and optical layers.
- Integration of ATM and IP networks.
- Traffic Engineering.
- Provisioning of traffic with different qualitative Care of Service (CoS).
- Provisioning of traffic with different quantitative Quality of Service (QoS).
- Providing IP based VPNs.

MPLS is expected to address scaling issues faced by the continuously growing Internet [3][4].

It does not require a specific label-distribution protocol. Protocols that can be used for this purpose are RSVP (Resource Reservation Protocol), BGP (Border gateway Protocol), LDP (Label Distribution Protocol), CR-LDP (Constraint Based Routed Label Distribution Protocol) and OSPF.

1.3 VoIP and VoIPoMPLS

Voice over Internet Protocol (VoIP) is used to transit voice over a network using IP. VoIP has three modes of operation namely Voice Directly Over IP, Voice Directly Over User Datagram Protocol (UDP) and Voice Directly Over RTP (Real-Time Protocol). Running voice directly over IP means direct placement of voice traffic into the user field of the IP datagram. UDP helps manage Internet port numbers between computers and applications. These ports identify a layer 7 application. Port numbers when concatenated with IP addresses form sockets which help in uniquely identifying end-point connections. They are also useful in identifying sessions between applications. Some call processing protocols like SIP (Session Initiation Protocol) cannot function effectively without the use of ports. RTP supports real-time traffic. It operates with both unicast and multicast applications. RTP is used for audio traffic identification like the ones encoded in G.723, G.729 etc. standards. Additional uses of RTP are sequence numbering, timestamping and delivery monitoring. Applications run RTP on top of UDP utilizing UDP's port multiplexing and checksum services.

In case of VoIPoMPLS the protocol stack contains voice data which is encapsulated in IP layer protocols like RTP/UDP/IP followed by MPLS protocol encapsulation. Header compression may be employed in some implementations. The result is then carried over any link layer protocol like FR, ATM, PPP or Ethernet. It is essentially a method of implementing VoIP which is supported by existing IETF (Internet Engineering Task Force) Standards.

1.4 Problem Statement

A major requirement in voice networks is high availability and reliability. Subscribers should be able to have uninterrupted access to services and should not suffer from dropped calls. Downtime should be kept to a minimum with backup resources taking over when any component (switch, link) fails. Voice and data networks are converging at a rapid pace requiring protocols, software and hardware that can guarantee high levels of availability.

QoS is the quality a customer can expect from a given service. It is a function of the Service Level Agreement. A number of factors are taken into consideration when specifying QoS that are

- **Latency –** It is the time between the sending and receiving of the packet. Response time is also related to latency and is the round trip time (twice the latency). It is an important element in IP Telephony.

- **Jitter –** The interarrival jitter is defined to be the mean deviation of the difference in packet spacing at the receiver compared to the sender for a pair of packets. The impact of jitter on real-time voice applications is significant.
- **Packet Loss –** It is the percentage of packet loss in the transmission. Tolerance for packet loss varies with different applications.
- **Throughput –** It is the amount of data transferred between two nodes in a given amount of time. In other words bandwidth has a considerable role to play in determining the QoS.

Telephony is expected to have an enormous growth on the Internet in the next few years. Time critical applications often use proprietary network standards. Future IP networks will be able to provide better bandwidth guarantees and real-time applications will be able to use IP as the common network platform.

VoIPoMPLS encapsulates voice samples as IP datagrams (e.g. RTP/UDP/IP) followed by encapsulation in the MPLS PDUs. Header compression can be utilized in some implementations. It is possible to compress the 44 bytes of VoIP header (RTP+UDP+IP) down to 4 or 2, which makes VoIPoMPLS a major competitor to all other voice transport technologies. The resultant packets are carried over an MPLS transport arrangement such as Frame Relay, ATM, PPP or Ethernet. MPLS bolsters VoIP through efficiency of header compression and scalability through flow aggregation.

Absence of empirical analysis to determine which technique (i.e. IP or MPLS) furnishes better performance is probably the reason why VoIPoMPLS has not been widely adopted by the Service Providers.

The MPLS network holds a lot of promise when it comes to high speed routing within a backbone network. Since the core routers are doing switching instead of routing the advantage gained over normal IP based forwarding has to be greater. As the need for efficient transport of real time traffic increases the need to find the perceived advantage MPLS based backbone network provides over normal IP forwarding in terms of reduced delay and jitter has become a necessity. The purpose of this study is to compare and contrast these two VoIP and VoIPoMPLS. In order to achieve the aforementioned objective, the following tasks need to be performed:

- Measurement-based comparison of VoIP and VoIPoMPLS to verify the known simulation-based results [5].
- Experimental setup of software routing using IPv4 and MPLS for measurement-based testing.
- Workload characterization and experimental design and excution using packetized voice traffic.

1.5 Main Contributions

The main contributions of this thesis are summarized as follows:

- Comparison of Router Performance across IPv4 and MPLS delay and jitter using high and low level priorities of voice traffic using the same testbed.

- Comparison of Router performance with and without network congestion across IPv4 and MPLS.
- Measurement based analysis of router CPU utilization, router interrupts/sec and per packet delay at a router.
- Comparison of simulation based results with measurement based results.

1.6 Organization of the Thesis

The rest of this thesis is organized as follows. Chapter 2 gives an introduction to the MPLS technology, architecture and functioning. In Chapter 3, we present a literature survey related to various areas of research in MPLS. Chapter 4 introduces the approach that was employed in the experimental study. Chapter 5 provides measurement based evaluation of results obtained. In chapter 6, we present the conclusion and future direction of work.

CHAPTER 2

2 BACKGROUND

2.1 MPLS

Multiprotocol label switching (MPLS) was developed to support different kinds of network protocols such as IP, ATM, Frame Relay and so on. A Label Switching Router (LSR) is a router that supports MPLS. An MPLS domain is made up of a group of LSRs with the same MPLS level. The edge router of an MPLS domain can be an ingress router or an egress router.

The ingress router of an MPLS domain analyzes the packet's network layer header and assigns the packet to a particular forwarding equivalent class (FEC). FEC is used for associating discrete packets with destination address and a class of traffic. It allows grouping of packets into classes. A label is used to identify the association of a packet to a particular FEC. Different FECs and their associated labels are used for different classes of service. Once the packet has been assigned to a particular FEC no further analysis of the packet's network layer header at subsequent hops within the same MPLS domain takes place. The label is used in each hop as an index to a table which specifies the next hop and a new label. MPLS domain can therefore be considered as label driven. The path through one or more LSRs followed by packets in a particular FEC is called a label switched path (LSP). A label associated with an FEC can change over an MPLS domain as long as each

router maintains a label mapping table so that the router can recognize to which FEC the incoming label is to be mapped. Only then can a new label be assigned and the packet can be forwarded to the next hop. The MPLS header of a packet is removed before it leaves the MPLS domain.

There has to be an agreement between neighboring routers before the arrival of a packet in order to set up a valid table for packet forwarding. If the router in an MPLS domain receives a packet with an empty label then the router will either drop the packet or will be forced to analyze the network layer header of the packet.

A labeled packet does not necessarily carry only a single label. If an MPLS domain is further subdivided into a number of sub domains then there has to exist a model in which a labeled packet carries a number of labels, organized as a last in first out stack. This is called the label stack. An MPLS network can also support a hierarchical architecture. The processing of a labeled packet is independent of the level of hierarchy. The top label is processed first regardless of the possibility that some other labels may have been above it in the past or below it at present.

An unlabeled packet is a packet whose label stack is empty. If a label stack has a depth of n labels then the label at the bottom of the stack is referred to as the level 1 label. The one above it is referred to as the level 2 label and the one at the top is referred to as the level n label.

The advantages that MPLS holds over the conventional connectionless forwarding are as follows:

1. MPLS routers do table lookup and label replacement instead of inspecting the packet's network layer headers.
2. In conventional forwarding the packets are memoryless about their ingress routers whereas the packets entering an MPLS network are distinguishable so that forwarding decisions that depend on the ingress router can be easily made.
3. The label table of an MPLS switch may contain information like the precedence or the class of service predetermined for the corresponding FEC. Conventional forwarding on the other hand considers information encoded in the packet header.
4. It is sometimes desirable to force a packet to follow a route that is explicitly chosen before the packet enters the network rather than the route be chosen by dynamic routing. This feature allows the implementation of traffic engineering. In conventional forwarding the packet carries an encoding of its route along with it. In MPLS, the association of the label with the table is used to represent the route in order to avoid carrying explicit route with the packet. This reduces the size of the packet overhead.

2.1.1 Basic Architecture

In order to avoid double assignment of any label value the binding of a particular label to a particular FEC is done by the LSR downstream. A sequence of label stack entries represents the label stack. Four octets represent a label stack entry. Each label stack entry contains a 20-bit label, a 3 bit experimental field, a 1 bit label stack indicator and an 8 bit TTL.

The label stack follows the data link layer header and precedes the network layer header. The top of the label stack appears earliest in the packet and the bottom appears latest. A number of label stack entries can be added in between the data link layer header and the network layer header that are popped out in First in First Out (FIFO) manner to determine routes for packets within the MPLS network. The network layer header follows the label stack entry which has the S-bit set. The S-bit is used to indicate the presence of label stack entries.

Each label stack entry consists of the following fields:

1. Bottom of stack (S): This bit is set for the bottom entry in the label stack and zero for all other label stack entries.
2. Time to live (TTL): A time to live value is encoded in this eight bit field. TTL value is set to the IP TTL value upon labeling and is decremented at each MPLS network hop. Upon popping of the last label off the stack, MPLS TTL is copied to the IP TTL field.
3. Experimental Use: The three bits in this field are reserved for experimental use.
4. Label Value: This field contains the value of the label. Upon receiving a packet the label at the top of the stack is looked up. Upon successful lookup the following information is obtained:
    a) The next hop to forward the packet to
    b) Operation to be performed on the label stack before forwarding like pushing, popping, or replacing labels.

The outgoing data link encapsulation may also be learnt in addition to learning the next hop and the label stack operation. Pushing multiple labels may cause the length of the frame to exceed the layer 2 MTU therefore the LSR must support maximum IP datagram size for labeling as a parameter, and any unlabeled datagrams greater in size than this parameter must be fragmented.

Selecting the LSP for a particular FEC is referred to as route selection. MPLS supports two options for route selection: (1) hop-by-hop routing and (2) explicit routing. Each node can independently choose the next hop for each FEC in hop-by-hop routing. In explicit routing the LSP ingress or the LSP egress specifies the LSRs in the LSP. The LSP is strictly explicitly routed if a single LSR specifies the entire LSP. If a single LSR specifies part of the LSP, the LSP is loosely explicitly routed.

The selection of explicitly routed LSP along with the sequence of LSRs may be done through configuration or dynamically by a single node. Policy routing or traffic engineering are the major reasons for doing explicit routing. Explicit routes need to be defined at the time of label assignment but they need not be specified with each IP packet. This makes MPLS explicit routing more efficient than the conventional IP source routing.

The penultimate hop popping is a scheme which requires an LSR to look up the label and knowing that it is the penultimate hop and what the egress hop is, pop the stack and forward the packet to the egress. The egress upon receiving the packet looks up the top label in order to make its own forwarding decision. This requires the penultimate and the

egress nodes to do a single table lookup. Penultimate hop popping can only be applied if it is requested by the egress and the penultimate node is capable of doing so.

For the purpose of traffic engineering, packets of a particular FEC are sometimes required to follow a specified route from an upstream router to a downstream router despite the probability that the downstream router may not be adjacent to either the upstream router or the destination. This concept is known as creating a tunnel the upstream router to the downstream router.

An LSP can be implemented as a tunnel and use label switching instead of network layer encapsulation to cause the packet to travel through the tunnel. Packets sent through an LSP tunnel constitute an FEC and each LSR in the tunnel assigns a label to that FEC. The transmit end point pushes a label for the tunnel on top of the label stack and sends the labeled packet to the next hop in the tunnel.

The next hop label forward entry (NHLFE) is used when forwarding a labeled packet. It consists of the following information:

1. Next hop of the packet
2. One of the following operations to be performed on the label stack:
    a) Replace the label at the top of the stack with a new label
    b) Pop the label stack
    c) Replace the label at the top of the stack with a new label and push one or more new labels onto the label stack
3. Class of service of the packet

If an LSR receives a packet whose next hop is the LSR itself then the LSR has to remove the label and make a forwarding decision based on the remaining part of the label stack. Such a situation will arise if the LSR is the egress.

The incoming label map is used to map each incoming label to a set of NHLFEs when forwarding labeled incoming packets. The FEC-to-NHLFE (FTN) is used to map each FEC to a set of NHLFEs when forwarding unlabeled incoming packets that are to be labeled before forwarding.

An LSR examines the top label of the label stack in order to forward a labeled packet. It uses the Incoming Label Map (ILM) to map the label to the NHLFE. The information in the NHLFE is used by the LSR to determine where the packet is to be forwarded next and performs an operation on the packet's label stack. A new label stack is then encoded into the packet and the result is forwarded by the LSR.

In order to forward an unlabeled packet the LSR analyzes the network layer header to determine the packet's FEC. It then uses the FTN to map this to an NHLFE. The information in the NHLFE is used by the LSR to forward the packet and then operates on the packet's label stack. The new label stack is then encoded into the packet and the result is forwarded.

The upstream or the downstream LSR can initiate the request for a label-FEC binding. The label assignment and distribution must be in a downstream-to-upstream direction to make sure that each LSR can uniquely interpret each incoming label.

2.1.2 Label Distribution

A set of procedures by which a downstream LSR informs its upstream peer of the label-FEC binding it has assigned is called a label distribution protocol. Two LSRs using a label distribution protocol to exchange label-FEC binding information are known as label distribution peers with respect to that binding.

The label distribution protocol may also encompass any negotiations in which two label distribution peers need to engage. A number of different label distribution protocols have been standardized. Protocols like BGP and RSVP have been extended in order to adapt to the label distribution.

The process in which an upstream LSR explicitly requests a label binding for a particular FEC from its next hop with respect to that FEC is called downstream-on-demand label distribution. It is also possible for a downstream LSR to distribute bindings to its upstream LSRs that have not explicitly requested them. This process is called unsolicited downstream label distribution.

The type of label distribution used may depend on which characteristics of the interfaces the implementation can support. Both techniques however can be implemented in the

same network at the same time as long as the label distribution peers have agreed on a particular type.

A downstream router may have distributed a label binding for a particular FEC to an upstream router even though it is not its next hop with respect to that FEC then the upstream router has alternative actions to take. The first is to maintain and keep track of such a binding known as liberal label retention mode and the second is to discard the binding known as conservative label retention mode.

In case of liberal label retention mode, the upstream router may immediately begin using the binding once the downstream router becomes its next hop for that FEC. On the other hand in case of conservative label retention mode the upstream router must reacquire the binding once the downstream router becomes its next hop.

Quicker adaptation to routing changes is possible through liberal label retention mode. Conservative label retention mode allows an LSR to maintain fewer labels.

Some FECs correspond to address prefixes that are distributed via a dynamic routing algorithm whose setup can be done in one of two ways, independent LSP control or ordered LSP control.

In independent LSP control, when an LSR recognizes a particular FEC, it makes an independent decision to bind a label to that FEC and distribute that binding to its label distribution peers.

In ordered LSP control, an LSR only binds a label to a particular FEC if it is the egress LSR for that FEC or if it has already received a label binding for that FEC from its next hop for that FEC. Ordered control is used where it is to be ensured that traffic in a particular FEC follows a path with some specified set of properties which is also the purpose of traffic engineering.

In independent control, some LSRs may begin label switching of traffic in the FEC even before the LSP is setup; which results in some traffic in the FEC following a path that does not have the specified set of properties. If the recognition of the FEC is a consequence of the setting up of the corresponding LSP, ordered control also needs to be used.

Both techniques, namely ordered control and independent control, are fully interoperable. If all LSRs in an LSP are not using ordered control the overall effect on network behavior is largely that of independent control as there is no guarantee that an LSP is not used until it is fully setup.

If two label distribution peers are interior gateway neighbors then they are referred to as local label distribution peers, otherwise they are called remote label distribution peers. An LSR performs label distribution with its local label distribution peer by sending label distribution protocol messages, which are addressed to the peer directly.

An LSR can perform label distribution with its remote label distribution peers in one of the following ways:

1. Explicit peering: In this method the LSR distributes labels to a peer by sending label distribution protocol messages that are addressed to the peer directly. This technique is useful when the number of remote label distribution peers is small or the number of higher level label bindings is large or the remote label distribution peers are in distinct routing areas or domains. It is apparent that the router needs to know which labels to distribute to which peers.
2. Implicit Peering: In this method the LSR distributes higher level labels to its remote label distribution peers, encodes a higher level label as an attribute of a lower level label and then distributes the lower level label along with this attribute to its local label distribution peers. This process continues till the information reaches the remote peer. This technique is useful when the number of remote label distribution peers is large. Implicit peering requires the intermediate nodes to store information that they might not be directly interested in.

MPLS can be implemented with different kinds of label distribution protocols. It does not specify a specific standard rule for choosing which label distribution protocol to use in which circumstances.

In some cases it may be required to bind labels to FECs that can be identified with routes to address prefixes. A widely deployed routing algorithm that distributes those routes can be used to piggyback label distribution on the distribution of the routes themselves.

BGP is used to distribute such routes. Using BGP to distribute labels to its BGP peers for label distribution has a number of advantages. BGP route reflectors are allowed to distribute labels providing a significant scalability advantage over using label distribution protocol between BGP peers.

Label Distribution Protocol (LDP) adopts the topology driven label assignment approach. In order to discover potential LDP peers, an LSR sends Hello messages over UDP periodically to its neighbors. On discovery of an LDP peer the LSR tries to establish a TCP connection to its peer. After the establishment of the TCP connection the two LSRs negotiate session parameters such as label distribution options, valid label ranges and valid timers. After the successful negotiation between two LSRs the establishment of an LDP session is complete. LDP messages are then exchanged over the LDP session. Some of the notable LDP messages are label request, mapping and label withdraw. An LSR uses a label request message to request a binding for an FEC from its peer. An LSR advertises FEC-label bindings to its peer using label mapping message. An LSR uses the label withdraw message to indicate to its peer to stop using FEC-label bindings that were previously advertised.

RSVP-TE (RSVP-Traffic Engineering) has also been specified as an extension to RSVP to establish traffic engineered LSPs. RSVP-TE adopts the request driven label assignment approach and allows an explicitly routed LSP between each LSR pair. In explicit routing the route taken by a packet is determined by a single node usually the ingress. A useful application of explicit routing is traffic engineering where maximization of network

resource utilization translates to effective mapping of traffic flows on the network topology.

RSVP-TE extends the RSVP Path message by including a label request object to request a label binding. The label binding is distributed upstream by extending the RSVP Resv message to include a label object in response to the request. RSVP-TE follows the downstream-on-demand label distribution mode. By including an explicit route object (ERO) in the Path message, explicit routing is implemented. The nodes along the explicit route are listed typically by the ERO. An LSP can be associated with a particular setup and holding priorities through the use of RSVP-TE. An LSP with a higher setup priority is allowed to preempt another LSP with a lower holding priority. An LSP with a lower holding priority is torn down if a particular link does not have sufficient bandwidth for a new LSP that has a higher setup priority. Bandwidth released by the LSP that was torn down can be used for setting up the new LSP.

For traffic engineering it is desirable to set up an explicitly routed path from ingress to egress. Applying resource reservation along the path is also desirable. There can be two approaches to this:

- An existing protocol used for setting up resource reservation can be extended to support explicit routing and label distribution.
- An existing protocol used for label distribution can be extended to support explicit routing and resource reservations.

A label distribution protocol is used between nodes of an MPLS network to establish and maintain the label bindings. For MPLS to operate correctly label distribution information needs to be transmitted reliably and the label distribution protocol messages pertaining to a particular FEC need to be transmitted in sequence. The capability to carry multiple messages in a single datagram and flow control is also desirable.

2.1.3 MPLS Support of Differentiated Services

DiffServ behavior aggregates can be mapped to MPLS by multiplying the number of DiffServ Behavior Aggregates (BAs) into the number of MPLS FECs to create a table of BA-FEC tupples. The mapping of BA-FEC tupples to LSPs can be done in three ways as described in [6].

- E-LSP: E-LSP stands for EXP-inferred Per Hop Behavior (PHB) scheduling class LSP. Map each FEC to an LSP, map Differentiated Services Code Point (DSCP) for all the BAs in this FEC to equivalent MPLS EXP values, mark EXP field in MPLS headers accordingly when encapsulating IP packets. No more than eight DSCPs can be accommodated since the EXP field can only support eight values. Further optimization can be done by moving the drop precedence consideration out of EXP field and mapping them to equivalent layer 2 functions. This reduces the number of DSCPs to be mapped to only six.
- L-LSP: L-LSP stands for Label only inferred PHB scheduling class LSP. Create an LSP for each BA-FEC tupple. Each LSP would have an equivalent traffic management profile to the DiffServ BA it is carrying. This causes the number of LSPs for a given FEC to multiply by the number of BAs. Optimization can be

done by using the EXP field to carry the drop precedence value of the DSCP, thus reducing the number of LSPs to one third. E.g. four LSPs are required to support the AF group type, instead of twelve.

- A hybrid of E-LSP and L-LSP can be used. E.g. it may be useful to implement L-LSP for Expedited Forwarding (EF) and E-LSP for Assured Forwarding (AF) and Best Effort (BE).

MPLS packets arriving at a core node cannot simply be forwarded by label swapping since DiffServ packets must be processed as aggregates on a per hop basis. Packets from all LSPs have to go through DiffServ PHB processing at each hop, where some packets may be associated with a lower PHB and must be relabeled accordingly. In order to preserve packet ordering the packets from the same BA must be forwarded on the same LSP. Even though extra processing is required for MPLS to support DiffServ, the model is still quite scalable as no state information need to be kept at each node about the traffic and the PHB processing can be scaled with processor and memory technology.

2.1.4 Applications of Multiprotocol Label Switching

The main idea of MPLS is the use of a forwarding paradigm based on label swapping that can be combined with a range of different control modules. Each module is supposed to assign and distribute a set of labels as well as maintain other relevant control information. An MPLS router may include

- A unicast routing module which builds up the routing table using the conventional IP routing protocol, assigns labels to the routes, distributes labels using the label distribution protocol (LDP), etc.

- A traffic engineering module which enables explicitly specified label switched paths to be set up through a network for traffic engineering purposes.
- A virtual private network (VPN) module which builds VPN specific routing tables using BGP and distributes labels corresponding to VPN routes.

MPLS allows forwarding of a packet regardless of the contents of the packet's IP header as it allows different modules to assign labels to packets using a variety of criteria. This property is highly essential for VPN and traffic engineering support.

2.2 Competitors of MPLS

This section focuses on other network technologies competing with MPLS in order to increase their stakes in the business of high speed communication networks. Network architectures such as Integrated Services, Differentiated Services and network technologies like ATM, Frame Relay, Point-to-Point Protocol and the upcoming Layer 2 Tunneling Protocol immediately come to mind. A brief introduction of these architectures and technologies will be followed by a comparison with MPLS over a wide range of parameters and factors to determine the most viable technology for the future.

2.2.1 Integrated Services (IntServ)

The IETF developed the Integrated Services model [7], which requires resources like bandwidth and buffers to be reserved for a given data flow to ensure QoS for the application that requested it. The model as shown in figure 2.1 makes use of packet classifiers to identify flows that are supposed to receive a certain level of service. Packet

schedulers are required to handle the forwarding of different packet flows in a manner that ensures that QoS commitments are met. Admission control is employed to determine if a router has the necessary resources to accept a new flow. It can be said that this model resembles that of the ATM where admission control along with policing are used to provide QoS to individual applications.

The Resource Reservation Protocol (RSVP) is used by this model to reserve resources along the traversed path by a new flow requesting a QoS service. RSVP informs each router of the requested QoS. When the flow is found admissible, each router adjusts its packet classifier and scheduler to handle the given packet flow.

The traffic and QoS requirements of a flow are described by a flow descriptor. Filter specification (filterspec) and flow specification (flowspec) make up the flow descriptor. The filterspec is used to identify packets that belong to the flow required by the packet classifier. The flowspec consists of a traffic specification (Tspec) and a service request specification (Rspec). Traffic behavior of the flow in terms of a token bucket is specified by the Tspec. The requested QoS in terms of bandwidth, packet delay or packet loss is specified by the Rspec. The Integrated Services model also introduces two new services: guaranteed service and controlled load service.

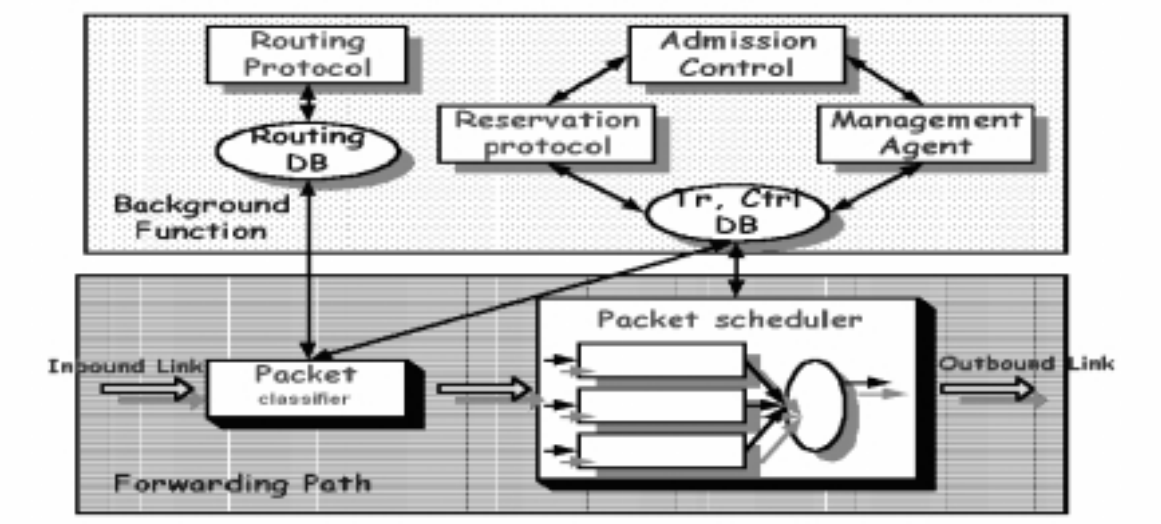


Figure 2.1 Router model in IntServ [7].

2.2.1.1 Guaranteed Service

Applications that require real-time service delivery, guaranteed service in the Internet can be used for them. For such applications data that is delivered after a certain time limit is considered worthless. Guaranteed service thus provides a bound on the end-to-end packet delay for a flow.

Each router must know the traffic characteristics of the flow and the desired service in order to support guaranteed service. The router uses admission control to determine whether a new flow should be accepted based on this information. After the acceptance of a new flow, the router polices the flow to ensure compliance with the promised traffic characteristics.

2.2.1.2 Controlled Load Service

This service is intended for adaptive applications that can tolerate some delay but are sensitive to traffic overload conditions. The performance of these applications deteriorates

with the increase in network load. This service was therefore designed to provide approximately the same service as the best effort service in a lightly loaded network regardless of the actual network condition. An application requesting a controlled load service can expect low packet loss, low queuing delay which is a typical behavior of a statistical multiplexer that is not congested. The controlled load service requires less implementation complexity than the guaranteed service requires.

An application requesting a controlled load service has to provide the network with the token bucket specification of its flow. The network makes use of policing and admission control to ensure that enough resources are available for the flow. Flows that conform to the token bucket specification are to be served with low delay and low loss. Non-conforming flows are to be treated as best-effort service.

2.2.1.3 Differences

IntServ places a heavy processing load on routers in the core of the network and is not quite scalable in large networks with many IntServ flows. The reason for it is that it operates at each level of the individual packet flow. This also gives an indication as to how the processing load is proportional to the number of IntServ flows. The amount of state information increases proportionally with the number of flows requiring extra storage space. This model also makes the router much more complex as it needs to implement the RSVP protocol, admission control, packet classifier and per flow packet scheduling algorithm. IntServ also lacks the functionality to aggregate flows into classes before they cross the network so as to reduce the processing load. IntServ requires continuous signaling because of its soft state architecture.

On the other hand in MPLS architecture the load on the core MPLS routers is reduced as all the extra processing overhead is taken care of by the ingress and egress routers. The core routers in fact do switching thus decreasing the delay and jitter which is so crucial for real time traffic. MPLS is highly scalable as it makes use of labels and label switched paths to aggregate traffic flows. It also does not require continuous signaling.

2.2.2 Differentiated Services (DiffServ)

IETF introduced the DiffServ model [8], which was intended to be simpler than IntServ and much more scalable. Scalability is achieved in two ways. First, per aggregate service replaces per flow service and second, complex processing is moved to the edge of the network from the core.

The DiffServ model aggregates the entire customer's requirement for QoS. A customer or organization has to have a Service Level Agreement (SLA) with its service provider if it wishes to receive differentiated services. SLA is a contract between a customer and a service provider that specifies the forwarding service that the customer will receive. An SLA includes a traffic conditioning agreement (TCA), which gives detailed service parameters such as service level, traffic profile, marking and shaping. An SLA can be static or dynamic. Static SLAs are negotiated on a long term basis between the customers and the service providers. A bandwidth broker protocol is used to effect SLA changes when dynamic SLAs (SLAs which change more frequently) are employed. Dynamic SLAs are tuned towards servicing customers with their dynamically changing service demands.

A customer has to mark its packets by assigning specific values in the type-of –service (TOS) field (renamed to the DS field) if he wishes to receive different service levels for different packets. Packet marking is usually done at the customer premises, at a host or at a customer's router. Different values of DS field correspond to different packet forwarding treatments at each router called the per hop behaviors (PHB). In the DiffServ model, shown in figure 2.2, the router reserves resources on an aggregate basis for each PHB.

The ingress router in the provider's network needs to support traffic classification and traffic conditioning to ensure that the traffic entering the service provider's network follows the rules specified in the TCA. Traffic classification is performed by a traffic classifier which matches the content of some portion of the packet header with some pre-defined sets so that packets can be directed to appropriate data paths inside a router to receive appropriate treatments. The content may come from the DS field, IP source address, IP destination address, protocol ID number, source port number, and destination port number. Traffic conditioning is performed by a traffic conditioner which combines elements such as metering, marking, shaping and dropping.

A DiffServ capable router relies on the value of the DS field of each packet and uses buffer management and scheduling mechanisms to deliver the specific PHB. The value of the DS field is set at network boundaries. In order to provide additional classes of service, the IETF has defined additional PHBs.

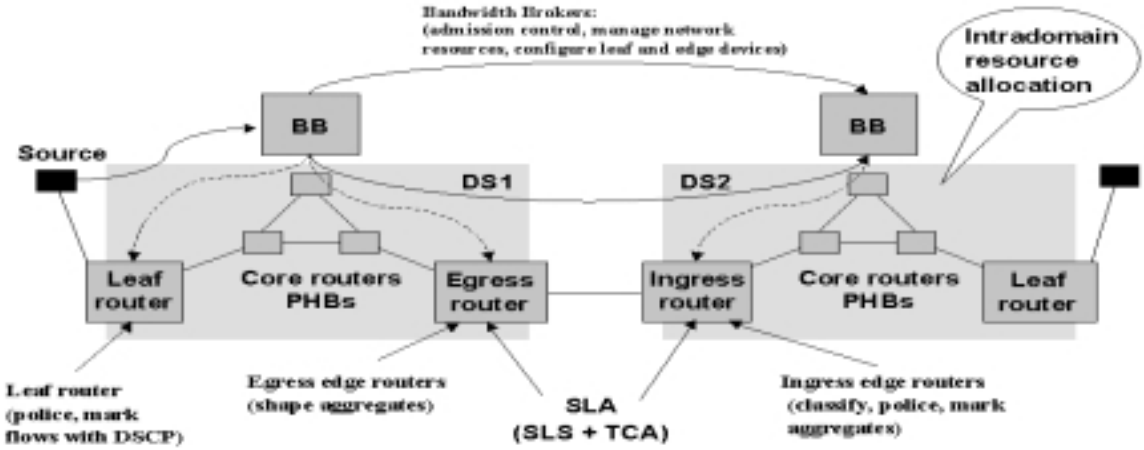


Figure 2.2 Main components of a DiffServ Network [9].

Two additional PHBs have been defined: the expedited forwarding PHB (EF PHB) and the assured forwarding PHB (AF PHB). EF PHB provides a low loss, low-latency, low jitter, assured bandwidth, end-to-end service through DS domains. The service received by the end hosts using EF PHB is analogous to a virtual leased line. AF PHB delivers the aggregate traffic from a particular customer with high assurance as long as the aggregate traffic does not exceed the traffic profile. The customer is allowed to send its traffic beyond the traffic profile under the condition that excess traffic may not be given high assurance. AF PHB is not intended for low-latency, low jitter applications.

DiffServ model is scalable but it does not guarantee service on end-to-end basis. DiffServ requires accurate link state (e.g. available bandwidth, packet loss rate, delay, etc.) and topology information. The time varying low capacity of the network make maintaining accurate routing information very difficult. Another important aspect is that there is only a statistical guarantee of a specific behavior. The statistics can be made arbitrarily good, but there is never an absolute guarantee. It is simply a mechanism to decide which packets to delay or drop at the expense of others in a situation where there is not enough network

capacity. Supposing that DiffServ is working by dropping packets selectively and the traffic in question is already close to saturation, then any increase in traffic will result in Bronze service being taken out altogether. As Internet traffic is bursty in nature, it is bound to happen on a regular basis when the traffic on a link nears the limit at which DiffServ becomes needed. Most ISPs employ DiffServ only as a way of customer network utilization to allow greater overlooking of their existing capacity.

MPLS is about routing (switching) while DiffServ is about queuing, scheduling and dropping. DiffServ is implemented in an all IP network i.e. it works at layer 3 whereas MPLS gives the flexibility of use with any layer 2 protocol like ATM, Frame Relay, PPP or Ethernet. It is very much a question of convenience when it comes to making a decision on which architecture to choose as both architectures are backed by big time players. CISCO for example is backing MPLS to furnish IP-routing functions for its career-class ATM equipment. However both these architectures are being used to complement each other whereby DiffServ Code Point (DSCP) is mapped onto LSPs to achieve greater QoS.

2.2.3 Frame Relay (FR)

Frame Relay is a high performance WAN protocol that works at the physical and data link layers of the OSI reference model. It was originally designed for use across the Integrated Services Digital Network interfaces. Nowadays it is used over a number of other network interfaces.

It is a packet switched technology. The network medium and the available bandwidth is dynamically shared between end stations in a packet switched network. Two techniques are used in packet switching technology:

- Variable length packets
- Statistical multiplexing

Efficient and flexible data transfer is made possible through the use of variable length packets. The packets are switched between various segments until the destination is reached.

The majority of Frame Relay traffic consists of TCP/IP or other protocols that provide their own flow control and error correction mechanisms. Frame relay does not care whether the frame it is switching is error free or not. It starts switching traffic as soon as it has read the first two bytes of addressing information at the beginning of the frame allowing the frame to end-to-end passing several switches and still arriving at its destination with only a few bytes delay. The delays are not noticeably different from direct leased line connections.

Packets are routed through one or more virtual circuits known as Data Link Connection Identifiers (DLCI). Most of the virtual circuits are Permanent Virtual Circuits (PVCs) which means that the network provider sets up all DLCI connections at subscription time. Switched Virtual Circuits (SVCs) on the other hand provide a link that only lasts as long as the session.

Frame relay implements two congestion notification mechanisms:

- Forward explicit congestion notification (FECN)
- Backward explicit congestion notification (BECN)

Both are controlled by a single bit contained in the frame relay frame header. Discard eligibility bit contained in the frame relay frame header is used to identify less important traffic that can be discarded during periods of congestion.

The Local Management Interface (LMI) is a set of enhancements to the basic frame relay specification. It provides extensions for managing complex Internetworks. These extensions include global addressing, virtual circuit status messages and multicasting. The LMI global addressing extension gives DLCI values global rather than local significance. DLCI values become DTE addresses that are unique in the frame relay WAN. LMI virtual circuit status messages provide synchronization and communication between frame relay DTE and DCE devices. These messages report on the status of PVCs.

Frame relay provides a cost effective way of providing a secure private IP based network. Frame Relay privacy is guaranteed by the nature of the network, backed up by legislation unlike the VPNs employed by the companies over the Internet for inter-company communications which exposes the organization to serious security issues like viruses and hackers.

Frame relay is also deployed as a low cost carrier to replace the network of leased lines previously used to connect ATM machines and other legacy devices to head office mainframes. Frame relay connections are also used for high-end Internet connections.

In a common frame relay network a T1 multiplexer is equipped with both frame relay and non-frame relay interfaces. Frame relay traffic is forwarded out the frame relay interface onto the data network whereas non-frame relay traffic is forwarded to the appropriate application or service such as Private Branch Exchange (PBX) for telephone service or to a video conferencing application.

Theoretically there is no data integrity, however by running an upper layer protocol above frame relay that is capable of recovering from errors such as TCP/IP, X.25 or IPX the data is delivered quite reliably.

There is also no true flow control; however it includes features designed to control and minimize frame loss at the user level. Network overhead is reduced by implementing simple congestion notification mechanisms.

Since frame relay routinely dumps error frames, in speech quality terms this can result in poor quality of voice transmission. It also does not support a number of traffic varieties.

On the other hand MPLS through traffic engineering supports different traffic varieties. It minimizes delay through fast switching in the core. It is a control plane technology to optimize network behaviors of mapping layer 3 end-to-end data flow to layer 2 traffic

between adjacent network nodes. It performs routing path control and connection control functions.

2.2.4 ATM

Asynchronous Transfer mode (ATM) is a method for multiplexing and switching that supports a broad range of services. It is a connection oriented packet switching technique that can provide QoS guarantees on a per connection basis.

ATM contains some desirable features of packet switching and time-division multiplexing (TDM) circuit switching. All information flows are converted into short 53 byte fixed length packets called cells. The 5 byte headers contained within the cells are essentially pointers to tables in the switches. ATM has some of the following features. Since it is packet based, it can easily handle services that generate information at variable bit rates. The fixed length cells and the abbreviated header facilitate hardware implementations that result in low delay and high speeds.

Information flows from a number of users are converted into cells and sent to an ATM multiplexer. The cells are then arranged into one or more queues and are scheduled. The scheduling strategy provides for the different qualities of service required by the different flows. For specific information flows, ATM does not reserve transmission slots, thus enabling it to have the efficiencies of packet multiplexing. The transmission of cells is not synchronized to any frame structure as in the case of TDM systems.

ATM networks require a connection setup prior to the transfer of cells and are connection oriented. The connection setup is analogous to that of the virtual circuit packet switching networks. The source provides a traffic descriptor that describes the manner in which cells are produced during an ATM connection setup e.g. peak cell rate in cells/seconds, sustainable cell rate in cells/second and maximum length of a burst of cells. The source also specifies a set of QoS parameters that the connection must satisfy e.g. cell delay, cell loss and cell delay jitter. Identification of a path through the network that can meet these requirements is part of the connection setup procedure. At every multiplexer along the path a connection admission control procedure is carried out. This path is called a virtual channel connection (VCC).

A chain of local identifiers establish the VCC and are defined during the connection setup at the input port to each switch between the source and the destination. ATM very strongly resembles virtual circuit packet switching. The major difference is the ATM's use of short, fixed length packets. The implementation of switches is simplified and high speed operation is possible through the use of this approach. ATM switches with capacities of up to 640 Gigabits/second have been deployed in the field. A finer degree of control over the scheduling of packet transmissions is achieved through the use of short fixed length packets. Shorter packets result in a smaller minimum waiting time until the transmission line becomes available for the next transmission.

ATM uses the concept of virtual path to allow the bundling of flows through a common path. VCCs can be bundled into an aggregate path called a virtual path connection (VPC). The purpose of the VPC is to switch only virtual paths and it happens to pass through an

ATM cross connect. Virtual Channel Identifiers (VCIs) are used to identify the VCCs within the VPC.

A virtual circuit in an ATM requires two levels of identifiers: an identifier for the VPC, the VPI and a local identifier for the VCC, called the virtual channel identifier (VCI). If VCCs are considered to be small pipes then the VPC can be thought of as a larger digital pipe that takes traffic streams from these pipes and handles them as a single unit. The cells belonging to a specific VCC are identified by a two part identifier that consists of a VPI and a VCI. VCCs bundled into the same virtual path have the same VPI and their cells are switched in the same manner over the entire length of the virtual path. Switching is based on the VPI only at all switches along the virtual path and the VCIs remain unchanged. The VCIs are used only at the end of the virtual path. The VCI/VPI structure can support a very large number of connections and hence provide scalability to very large networks.

ATM was supposed to provide high speed high density layer 2 services by exposing internal switch design into protocol. However, it was unable to deliver because of bad selection of cell size and selection of one to many cell packet relationships. ATM cells are fixed in size at 53 bytes of which 5 bytes are header leaving 48 bytes for user payload. This is a 9.4% overhead penalty. In addition to the 5 bytes of header, two additional bytes are used for error checking in some cases. Some cells are used to carry administrative data only [10]. This can result in 20 percent consumption of throughput on a given circuit. ATM is inefficient for long distance data transmission as data has to be repeatedly broken down into 48 byte payloads. A packet could be divided into a number of cells but a cell

could not contain two or more packets. Small sized headers did not allow the implementation of advanced services like ABR/fault tolerance/VPN etc. The one to many cell packet relationship imposed significant cell tax. Switch design was expensive and put a heavy toll on advanced service implementations due to high event rate, all because of the small cell size.

ATM results in a lot of bandwidth wasting when employed between different L2 networks. Consider for example the carrying of an IP packet – ATM header, ethernet header, IP header and customer traffic. This is the case when the packet has to traverse an ATM backbone network in between two Ethernet networks. On the other hand MPLS has one or two labels and IP packet. Managing such traffic is easy. An MPLS backbone network will be cheaper in the longer run as it will reduce operational cost and capital investment costs as MPLS equipment becomes cheaper.

The need to implement segmentation and reassembly for ATM at speeds beyond OC-48 also proved challenging and unnecessary given the much lower packet transmission times inherent at such high transmission speeds.

The most significant problem with IP over ATM overlay model is the need to build and manage two networks with dissimilar technologies. The overlay model also increases the complexity of the network architecture and the network design. Reliability is also an issue because more network elements now exist in series on the routed path. There is also a possibility of routing instability in the IP domain as a consequence of multiple PVC failures following a single interswitch link impairment in the ATM core.

2.3 Voice Transport Technologies

2.3.1 VoIP in the Internet and in Private Internets

The deployment of synchronous traffic over a private asynchronous Internet offers the same challenges as the public Internet. There is one difference between IP telephony in a public Internet and in a private Internet. An Internet can be designed to be much more cooperative than the Internet. Private networks can be more readily tuned than the Public Internet. That is why they support VoIP better than the public Internet.

2.3.2 Accommodating Voice and Data Requirements in a Network

Voice transmissions exhibit high tolerance for errors. Even if a voice packet gets distorted, the fidelity of the voice reproduction is not severely affected. Data packets however have a low tolerance for errors; one corrupted bit changes the meaning of the data. Voice packets can afford to be lost or discarded. If excessive delays are encountered in the network then the packets may be discarded because they are of no use if they arrive at the receiver too late. Voice fidelity is not severely affected if the lost packets are less than approximately 5% of the total packets transmitted [1]. On the other hand data packets can ill afford to be lost or discarded.

Another difference between voice and data transmission deals with network delay. To translate packetized voice to an analog signal in a real time mode, the two way delay for voice packets must be constant and generally must be less than 300 ms.

The network delay can vary considerably. The packets can be transmitted asynchronously through the network without regard to timing arrangements between the sender and the receiver.

Voice packets require short queue length at the network nodes to reduce delay or to make the delay more predictable. The short voice packet queues can experience occasional overflow resulting in packet loss. Data packets require longer queue length to prevent packet loss in overflow conditions.

2.3.3 Voice over IP, Voice over UDP or Voice over RTP

Voice can run directly over IP, over UDP then IP, or over the RTP then UDP and then IP. Running voice directly over IP means placing voice traffic directly into the user field of the IP datagram. UDP has long been a mainstay in an Internet. It is useful for VoIP operations as it helps manage Internet port numbers between computers and applications.

The UDP header contains the Internet source and destination port numbers that are required for proper execution of the layer 7 protocols. A port identifier is used to identify a UDP upper layer user in a host machine. Port number concatenates with an IP address to form a socket. The address has to be unique throughout the Internet and a pair of sockets uniquely identifies each end-point connection. Since port numbers used by different PCs can be the same, sockets provide a way of identifying traffic to and for a specific host.

Internet publishes reserved port numbers for frequently used higher level processes called well known ports. Sockets are used to identify the sessions between applications. Some VoIP based call processing protocols cannot function effectively without the use of ports.

RTP is designed to support real time traffic i.e. traffic that requires playback at the receiving application in a time sensitive mode such as for voice and video systems. RTP can operate with both unicast and multicast operations. RTP provides services such as payload type identification, sequence numbering, timestamping and delivery monitoring. RTP is usually run on top of UDP by applications to make use of UDP's port multiplexing and checksum services. If the underlying network provides multicast distribution then RTP can be used to transfer data to multiple destinations. The sequence numbers in RTP allow the receiver to reconstruct the sender's packet sequence which can be used to determine the proper location of a packet.

2.3.4 Voice Coder Functions

The main function of a voice coder is to encode pulse code modulation (PCM) user speech samples into a small number of bits in such a manner that the speech is robust in the presence of link errors, jittery networks and bursty transmissions. The frames are decoded back to the PCM speech samples and then converted to the waveform at the receiver. Some of the voice coders used today are G.711, G.726, G.728, G.729 and G.723.

2.3.5 Bandwidth Conservation with Voice Activity Detection

Voice Activity Detection is used to cease the sending of packets when voice level activity falls below a threshold. While reducing bandwidth consumption this tool can be a bit

tricky to implement. It can lead to clipping in which part of the speech is truncated and not carried in the VoIP packets. An approach to combat clipping is to continue to code and sample the speech pattern and allocate the packet to the samples but drop the packet if the voice energy does not meet a minimum threshold during an allotted time.

2.3.6 Constraint-Based Routing with MPLS and OSPF

Policy based routing when used with MPLS is called constrained routing or constraint-based (CR). Traffic engineering requirements for MPLS networks are met using this mechanism. CR can be set up as an end-to-end operation i.e. from ingress to egress. The ingress node initiates CR and all the affected nodes reserve resources. Constraint means that for every node in the network there exists a set of constraints that must be satisfied for the link or links between the two nodes. An example of a constraint is a path with minimum amount of bandwidth. Another example can be of a path that is secure.

Modified OSPF can be used to find such paths and is constrained to advertise paths in the routing domain that satisfy these kinds of constraints. CR attempts to meet a set of constraints and at the same time optimize some scalar metric. One such scalar metric is hop count for delay sensitive traffic such as VoIP. It is also a known fact that extra hops create jitter especially if the Internet is busy and the routers are processing a lot of traffic.

2.4 Software Routers

Software-based routers continue to be important due to the ease with which they can be programmed to perform a number of tasks. Pressure to extend the set of functions that routers support is taking place in several different areas:

- Routers at the edge of the Internet are programmed to filter packets, translate addresses, make level-n routing decisions, translate between different QoS reservations, thin data streams (limited bandwidth usage), and run proxy code.
- A new market in home routers is emerging, where in addition to running firewall and NAT code, the router is supporting functionalities that cannot be supported on computationally-weak consumer electronics devices.
- The distinction between routers and servers is blurring as routers that sit in front of clusters run application specific code to determine how to dispatch packets to the most appropriate node.
- At the fringe, the active network research community is designing an architecture that will allow future generations of routers to run arbitrary code, thereby enabling the deployment of application-specific virtual networks.

2.4.1 Performance vs. Functionality

In order to add advanced functionality to software based router, performance has to be compromised in one way or the other. E.g. in the presence of static routes routing is faster as route discovery and learning is eliminated. On the other hand dynamic routing requires the discovery of neighboring routers and the formation of a dynamic routing table that can delete aging entries when not used for some time.

Similarly, the addition of MPLS based routing functionality to an existing IP based software router requires either a kernel level or a driver level enhancement. The driver level enhancement becomes vendor specific even though it promises high speed functionality. However software based routers are pretty much there to support a variety of NICs. Therefore, kernel level improvement is generally employed. With the addition of code to the kernel, the processing overhead increases as the number of interrupts per second, CPU and memory utilization also proportionally increase. The effect of a specific functionality on the performance metrics of a software router will be part of this study.

2.4.2 Differences between IP and MPLS based Software Routers

In general if static routing is used an IP based router can be configured to perform simple forwarding decisions based on the longest prefix match. However if routing decisions have to be made based on SLAs then dynamic routing protocols have to be employed in order to provide customer specific services.

MPLS based software routers can also be used for forwarding with the help of LSPs that are statically setup between ingress and egress to provide services to known traffic along specific paths. If MPLS routers have to support dynamic SLAs then already existing dynamic routing protocols with extensions to fulfill the requirements of an MPLS network can be employed for dynamic path selection, bandwidth management or congestion control. Examples of such routing protocols are BGP and OSPF. MPLS also supports protocols which were written for it only, like Label Distribution Protocol (LDP) and Constraint based Routed LDP (CR-LDP) to support dynamic label distribution.

CHAPTER 3

3 LITERATURE REVIEW

3.1 Introduction

In this chapter we will divide MPLS related issues into distinct categories and examine the work that has been done in those areas, the level of performance achieved, their practical value and some of the open questions that remain unanswered. Those distinct areas are Differentiated Services/Integrated Services support of MPLS, QoS in MPLS networks, mechanisms for improved performance of MPLS networks and various MPLS based implementations.

3.2 Differentiated Services/Integrated Services support of MPLS

The Differentiated Service (DiffServ or DS) architecture classifies packets into aggregated flows or service classes that specify a forwarding mechanism or Per Hop Behavior (PHB) [11] whereas Integrated Services (IntServ or IS) architecture defines QoS and reservation parameters to obtain required QoS for an Internet flow [12]. DRUM [11] proposed an architecture that introduced Gold, Silver, Bronze and Best Effort (BE) service classes for transporting network traffic according to user requirements. The proposed architecture was tested on a simulator with limited number of nodes and

heterogeneity of the nodes was not mentioned or considered, also what effect will link failures have on the suggested architecture remains unanswered.

[13] tried to predict various methods for the implementation of DS and IS in IP/MPLS based Access Networks. Various methods had been proposed but how network components will be configured to perform the desired tasks is not mentioned. One major assumption made was the use of copper as the dominant media in the access network however with the introduction of 10 and 100 Gbps Ethernet in the market the hypothesis does not match the reality.

[14] enhanced E-LSP and L-LSP described in [15] and [16] respectively with per class traffic engineering (TE). They built on their previously proposed algorithm [17] that set up LSPs with delay constraint and bandwidth guarantees. Again the results are based on a size-restricted network that has been simulated and the time span for the simulation is too small. The graphs do not represent flow behavior in the situation of a link failure and the need for computations and signaling increased from the ones done by the original E-LSP.

[18] suggested the hardware structure of a DS Router that performed high speed switching by using hypothetical input queue and Virtual Output Queue (VOQ) however memory requirements, processing and signaling overheads are not discussed.

[19] proposed a policy based QoS management architecture for an MPLS network that supports DS. They have extended the COPS model presented in [20]. How management directives from QoS management system were encapsulated using CORBA and what problems were encountered in their encapsulation are not mentioned. How multi-vendor routers will affect the setup of such an architecture is not known and performance results have not been provided in order to judge its capability.

[21] provided a framework for supporting DS based end-to-end QoS in the Internet on MPLS based LSPs. Details regarding the mapping of the architecture on a real network are missing, and the results of the proposed architecture have not been presented based on real time traffic. The TEQUILA [21] functional architecture is too complex with too many parameters, and how this will translate onto a real network is the big question.

[12] presented a method of integrating IS based QoS in MPLS domains. Implementation of the proposed techniques on a network has not been provided. The author has accepted the fact that the proposed technique can flood the MPLS core with CR-LDP messages, which set up and teardown per flow CR-LSP. Also what effect will extra TLVs have on the memory requirements of the Routers have not been tackled. Finally, mechanisms have suggested overcoming the extra overhead generated by RSVP protocol.

3.3 QoSR of MPLS

[22] proposed a model which incorporates TE and QoS Routing (QoSR) of MPLS to support Internet Based Distance Learning (I-DL). The study was done on simulated networks where LANs loaded TransitNets with 33% of their total capacities, which is hypothetical at best and can vary in a real scenario. TransitNets can have varying number of resources like available bandwidth, link speeds and number of hops. This can directly effect the transit time of delay sensitive traffic.

[23] put forth a technique to perform TE for Resilience Differentiated QoS in MPLS networks. It has not been mentioned as to how the Network Management System calculates resources allocated for the restoration of Resilience Class 2 (RC2) demands offline. There is no mention of the reason as to why the ratio of multiple resilience classes was taken to be the way they were and graphs for the scenarios with no reserved resources, full restoration and full protection have not been provided. All schemes more or less have been found to make some compromises at the expense of other.

[24] tried to bring together the IETF-defined MPLS mechanisms with the MPLS forum-defined MPLS User to Network Interface in order to provide a complete picture of their implementation for a complete end-to-end QoS architecture.

[25] proposed the use of modules to provide transparent QoS protection for enhanced MPLS QoS routing. They have used protection need as a parameter in their consideration,

which is selected according to the administrator's experience, but a general idea as to what those needs can be is not mentioned. Protection for all the links is not present which can lead to a longer path for a packet in case of a link failure.

[26] presented an algorithm to overcome the problem of traffic flow blocking after the setup of an LSP. The algorithm works in two phases, off-line and on-line, however with the constantly changing demands of the customer how many times the network will have to undergo off-line setup is untested. Test results of the proposed algorithm have not been provided and the problem of link failures has not been addressed.

3.4 Mechanisms for improved performance of MPLS networks

[27] presented a pre-qualified recovery mechanism, which optimizes network performance by considering link usage. What has not been mentioned is the mechanism to handle prioritized traffic. The proposed mechanism also requires extensions to current IGP protocols for exchange of network performance information.

[28] put forth an active traffic and congestion control mechanism to maximize throughput and avoid congestion. The packet forwarding rates of routers have not been mentioned. The reason for bulk throughput being greater than MPLS Active Traffic and Congestion Control (ATCC) when delay is of 50ms has not been stated. The proposed mechanism

introduces traffic control mechanisms in the core of the MPLS network, which is against the original MPLS concept. How the mechanism will react to failure on a downstream link when a congestion notification is traveling on a Reverse Network Tree (RNT) is not mentioned.

[29] introduced a path protection mechanism for efficient and fast notification of faults in the MPLS network. They have not given any details on the performance of their proposed approach neither have they considered multiple link failures. Bandwidth constraints have also not been taken into consideration.

[30] developed a method of rerouting IP traffic through the use of MPLS bypass tunnels. The algorithm took into consideration only one link weight change at a time and shortest path computation. It must be realized that shortest paths are not necessarily the optimal paths.

[31] introduced partial and full ingress failure recovery mechanisms in MPLS networks. The schemes have not been implemented on a network. What effect will Control Plane Identity (CPI) have on the memory requirements and processing speed of the ingress node is not touched. How will the ingress react to inefficient use of link to its immediate downstream LSR is a point which has not been considered. Finally, what will be the criteria for use of pre-determined ingress in case of multiple options.

[32] gave details of a network architecture to provide TE in a SIP over MPLS based Network. The functionality of the policy server within the Access network has not been discussed. There are no indications of implementation of the proposed architecture on a test bed. Future work can be done in ensuring path protection of such a network.

[33] floated the idea of implementing data labeling in MPLS networks within layer 2 of the OSI model. How their implementation helped in bandwidth management has not been discussed in detail. Information on whether the modules generated extra processing overhead has not been indicated and how the MAC module will be able to push and pop labels at run time is unknown.

[34] brought into light a mechanism for reducing packet reordering and packet losses along with reduced overhead in MPLS network. The variables α and β used in the calculation of estimated traffic rate and estimated average length of a packet is not clearly explained and is hypothetical. The experimental results on other traffic types like Variable Bit Rate (VBR) and bandwidth constraint routing can be the focus for future work.

[35] unveiled their idea of caching and aggregation to reduce exchange of CR-LDP signals between an Ingress and Egress routers within an MPLS domain. Relevant background information regarding Type Length Value (TLV) is missing. The paper discusses what can be achieved through caching and aggregation but does not give details of how the Ingress router performs caching and aggregation. Their aggregation

mechanism is similar to the use of DS over an MPLS network. A variable related to priority-based traffic can be added to enhance the capability of the proposed schemes.

All the above mentioned proposals to improve the performance of MPLS networks helped us a great deal in understanding how different elements in the MPLS network function in the presence of an active traffic flow and how throughput is effected due to congestion. Also the relation of overheads to packet losses was found.

3.5 Various MPLS based implementations

RATES [36] is a server designed for efficient management of an MPLS network for setting up of LSPs using policy and bandwidth as core parameters. Specific policies that are handled by the server remain opaque. Implementation of the proposed server in a live network has not been discussed. How the server will react to different types of traffic like delay sensitive real-time traffic or bulk traffic is a point not provided in the description. Having a separate server for communication with the Ingress router can produce significant communication overhead, which can directly affect the processing rate of the Ingress Router.

[37] presented a three level architecture for a policy based management of an MPLS network, which can work independently (i.e. without continuous monitoring by the administrator) and dependently (i.e. with the control provided by the administrator). Detailed description of the implementation of network level policy is missing from the

text. What effect can relational database have in the policing of the Policy Enforcing Points can be the area of future work. Some major shortcomings of the research were the lack of constructs for defining an execution behavior, lack of conflict detection and conditions not being general.

[38] described two Layer 2 over MPLS solutions. One provided by Juniper Networks in the form of Circuit Cross Connect that mapped inbound L2 circuit identifier to an outbound LSP. One LSP per direction was required. For establishment of N circuits, 2N LSP tunnels were required that turned out to be a configuration burden. Another solution was provided in the form of Draft-Martini, which could employ any existing LDP for packet switching. It used a control word to preserve circuit sequentiality and LDP signaling with extended signaling between Provider Edge devices to establish an LDP session between two devices that are not directly connected.

[39] gave an idea of an MPLS based Internet Exchange. The focus of the proposed exchange was the proper functioning. However parameters like bandwidth constraints, priority based traffic and different types of traffic (EF, AF) were not given any consideration. A comparative statistical data based on performance with other existing architectures can be added to extend the study. Their future work will encompass areas like stability and reliability.

Wise<TE> [40] was a server designed for TE of large-scale MPLS networks. The functionalities of Configuration Package, Measurement Package and Miscellaneous Package within the Common Service interface have not been provided. The purpose of Automated Computing Environment Command Line Interface (ACE CLI) has not been defined. There is no feasibility study on the point of installation of the server in the network i.e. whether communication with Ingress Routers or with core Routers will be more beneficial. Target specific enforcement of policy can be introduced into the architecture. The role of Routing Advisor for Traffic Engineering (RATE) can be given to a module within the Wise<TE> server, which will contain alternate path protection strategies that can be implemented at run-time.

An MPLS based load-balancing architecture for web switching was given in [41]. The tests were carried out on large files but the implementation results with streaming media, search engine requests, application requests, distributed database access requests and e-commerce based service requests were not provided. Load balancing is being done at the dispatcher but there is also a need for load balancing at the LERs as one LER can be overwhelmed with HTTP requests while the other remains underutilized. The architecture can be improved to provide bandwidth guaranteed delivery and also provide path protection mechanisms. The LERs can be configured according to client request patterns to produce more efficient results. Adaptive load balancing and development of a queuing model will be the focus of future work.

An abstract implementation and information model of MPLS-TE was presented in [42]. Where prioritized traffic and bandwidth constraint fit into the model has not been mentioned. Parameters related to policy have not been considered. The model has to be enhanced to accommodate real time collection of data. The proposed model has been implemented on Wise<TE> and direction of future work is its implementation with a larger test-bed of network nodes.

All these MPLS based implementations helped us gain an insight into how LSPs were setup, load balancing is carried out and what happens at the Ingress, Egress and LSR routers in overloaded conditions.

3.6 VoMPLS and VoIPoMPLS

An approach, known as VoMPLS [43], encapsulates voice samples in MPLS protocol on top of an MPLS transport arrangement such as Frame Relay, ATM, PPP or Ethernet. VoMPLSoPPP has been found to be highly efficient with respect to bandwidth utilization in backbone networks. Argument in favor of VoMPLS suggests the use of VPNs and use of MPLS in backbone networks to provide seamless end to end VoMPLS for customers. However, the idea of using MPLS as opposed to IP in access networks may suffer from actual deployment constraints.

The simulation based experiments carried out in [5] consider voice traffic produced with Constant Bit Rate (CBR) and Available Bit Rate (ABR) which do not emulate the behavior of voice traffic in the Internet.

Our approach was contrary to the one mentioned in [43] where the experiments were conducted with voice samples that were encapsulated in the MPLS protocol on top of any L2 protocol like Frame Relay, ATM, PPP or Ethernet. However, this study gave us the idea of comparing performances of MPLS and IP networks with voice traffic that was basically packetized voice. We extended the work done in [5] by carrying out experiments on a testbed network with a more realistic Internet based voice traffic model.

CHAPTER 4

4 THE APPROACH

4.1 Linux Based Software Router

Linux routers are inexpensive, and can be configured as WAN-routers capable of running several different protocols. General-purpose computing hardware costs are significantly lower than special-purpose routing hardware. The major advantage of such routers is that Linux runs on low cost and widely supported hardware. Some of the reasons for using Linux routers are listed below:

- Linux routers are flexible as they allow running of higher-layer applications, such as firewalls and secure services.
- Linux routers are stable as their TCP/IP protocol stack has been reviewed by literally thousands of programmers.
- Linux routers are easy to administer.
- Linux routers are based on a widely available technology. As system hardware and adapters are being produced for an enormous market, costs are low and time to market cycles are short.
- Linux routers provide investment protection as vendors generally phase out a product line.
- Linux routers are expandable as there is no need to worry about the router chassis being able to support additional network adapters.

- Linux routers are adaptable to changing network technologies.

4.1.1 Enabling IP Forwarding on a Linux Machine

In order to enable IP forwarding on a Linux machine it has to have PCI or ISA slots to support multiple NICs. Then it has to be ensured whether the NICs have their driver support in Linux. The drivers of the NICs are either dynamically loaded as modules or hard-coded into the kernel. Once a NIC is detected by the Linux Operating System IP addresses have to be assigned to them and they have to be administratively put in active mode.

The next step is to ensure that programs like iproute2 and iptables have been added as modules or made part of the kernel. iproute2 allows defining of static paths whereas iptables helps define forwarding rules for specific types of traffic. iptables can be used as well to drop packets that do not conform to a defined criteria. iptables are employed to filter traffic based on prefix matches, port numbers, protocols, source and destination addresses.

4.1.2 MPLS Forwarding in Linux

The processing of MPLS packets and their label stacks is done through instructions. These instructions are modified by adding in and out labels and switch paths. The instructions can be overridden which can be helpful if we want to receive packets with a label stack of size greater than 1. Both incoming and outgoing labels have their own set of instructions. The list of MPLS instructions are shown in Table 4.1.

Table 4. 1 List of MPLS instructions.

Instructions	Brief Description
Pop	remove the top label from the label stack. (in/out)
Peek	look at the label on top of the label stack, look the label up in the list of incoming label for this interface, and start executing the instructions associated with it. If there is not a label, execute a dlv. (in)
Push	push another label on the label stack. (in/out)
Dlv	send this packet to the layer 3 protocol stored with this in label. (in)
Fwd	send a packet to an outgoing label structure to be processed (in/out)
Nffwd	mask:nf:key:nf:key
Dsfwd	mask:ds:key:ds:key
Expfwd	exp:key:exp:key
set_ds	ds
set_tc	tc
set_exp	exp
Set	last step before transmitting a MPLS packet. It copies the outgoing interface and the next hop layer 2 destination from the out going label structure. (out) set the incoming interface to something different then the REAL incoming interface (in)
exp2tc	exp:tc:exp:tc
exp2ds	exp:tc:exp:ds
nf2exp	mask:nf:exp:nf:exp
tc2exp	mask:tc:exp:tcp:exp
ds2exp	mask:ds:exp:ds:exp

The instructions shown in Table 4.1 are explained in [44]. A few of them were implemented at the Ingress, LSR and Egress routers. The format for incoming instructions is listed below:

pop

push:<label type>:<value>
dlv
peek
set:<interface>:<nh famliy>:<next hop>
fwd:<key>
Format for outgoing instructions is listed below:
push:<label type>:<value>
set:<interface>:<nh famliy>:<next hop>
fwd:<key>

'push' is used to push any label of our choice. They do not have to exist in any table and neither do they have to correspond to the label structure being operated on.

4.2 Testbed

The whole testbed is shown in figure 4.1. The experimental testbed consisted of 11 PCs. Four PCs were used for sending and receiving voice traffic. One PC was used to act as an NTP (Network Time Protocol) server for the four senders and receivers in order to determine accurate delay and jitter up to a fraction of a millisecond. The PC based routers were all PIIIs with 600 MHz CPU, 128 MB RAM running RED HAT Linux 9.0 with kernel 2.6.1. An MPLS patch [44] was used to patch up the kernel along with iptable and iproute files in order to provide MPLS based routing capability to the PC based routers. One PC acted as Ingress and one acted as Egress. The other four PCs were made as LSRs. The traffic was produced in point-to-point and point-to-multipoint fashion. The types of traffic produced were UDP based data traffic, UDP based voice traffic and TCP based background traffic. The software used to generate voice traffic as well as TCP and UDP based traffic was D-ITG (Distributed Internet Traffic Generator). In case of point-to-point

flow of traffic, a single path in the core of the network was used whereas both paths were used in point-to-multipoint traffic flow.

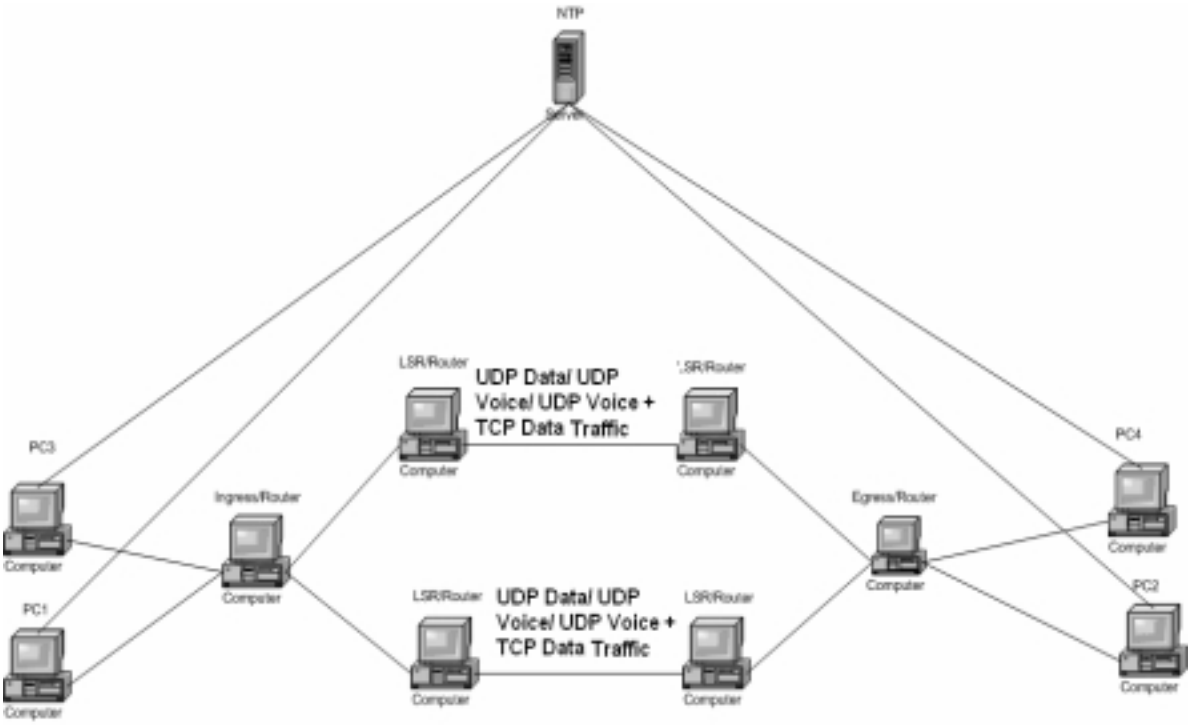


Figure 4. 1 Experimental Testbed Network.

4.2.1 Topology Selection

The main criteria for selection of the topology shown in figure 4.1 was to determine delay jitter, drop rate etc for voice, non-voice and mix traffic in both point-to-point and point-to-multipoint fashion. The two LSRs in each distinct path between the Ingress and the Egress were used to find the extent of switching advantage gained by the LSR over normal IP based Forwarding. Also the delay for voice traffic accompanied by TCP based background traffic was to be noted in order to make a comparison based on delay and jitter between MPLS based routing/switching and IP based forwarding of voice traffic.

This could give us an idea as to the benefit gained through the use of MPLS over IP. The NTP server made sure that the results were accurate when determining end-to-end delay and jitter for voice traffic.

4.2.2 Linux Kernel Patching

The Linux kernel patched with the MPLS source code consists of a label stack and data structures whose functionality is given below.

4.2.2.1 MPLS Label Stacking

The MPLS header is called a shim header. The label edge router or ingress is capable of adding multiple shim headers, called label stacking. The stack of shims is treated like a stack data structure. A POP represents the removal of a shim header showing another shim header or revealing the layer 3 header determined by the S bit. A PUSH adds a shim header on top of the stack of shim headers or the layer 3 header. Thus label swapping can be defined as a POP followed by a PUSH. A packet may sometimes be tunneled across an MPLS network. That requires a shim to be added on top of a previous one. This will produce a label stack of size 2. The process can be done by a single LSR or multiple LSRs. A labeled packet has a label stack of size 1. With every addition of a label the stack size increases.

4.2.2.2 MPLS Data Structures

Data structures are required to interpret and process labels. There are in general three such data structures, a data structure to interpret incoming labels at the LSR or the ingress, another one to add outgoing labels at the LSR and the last one to figure out the label to be added to a packet used by the ingress.

The data structure that interprets incoming labels is called ILM (Incoming Label Map). It consists of all the labels that an LSR or egress will recognize. An ILM entry consists of a label, opcode, FEC and an optional link to an outgoing data structure.

An incoming label is processed as follows

- Label extraction from top of the shim
- Label lookup in the ILM table
- Further packet processing based on the opcode

Each logical interface stores its own ILM table. MPLS packets arriving via that interface do label lookups for the ILM table of those interfaces. The opcodes in the ILM table are shown in Table 4.2.

Table 4. 2 opcodes in the ILM Table.

OPCODES	FUNCTION
POP_AND_LOOKUP	If the top shim has the S bit on: Extract the protocol type from the ILM POP the top shim Copy the TTL to the layer 3 header Using the protocol type, do a lookup on The layer 3 header that is exposed Else POP the top shim Extract the label from the shim that is exposed Extract the S bit Extract the EXP Extract label and create ILM Index Using the ILM Index Lookup the ILM Entry Execute the opcode in the ILM Entry End
POP_AND_FORWARD	Extract the outgoing route entry from the ILM POP the top shim If the outgoing route entry is a layer 3 route entry

	copy TTL to layer 3 header Using the outgoing route entry forward the packet to the outgoing interface
NO_POP_AND_FORWARD	Extract the outgoing route entry from the ILM Using the outgoing route entry forward the packet to the outgoing interface
SEND_TO_RP	Send the entire packet to the Route Processor

The second data structure that assists with outgoing labeling is the NHLFE (Next Hop Label Forwarding Entry). It consists of all the labels that can be pushed onto the packets. Each NHLFE contains a label, an outgoing interface and nexthop information. Packet processing through NHLFE goes through the following steps:

- New shim formation containing the label
- Pushing of shim onto the packet
- Packet forwarding to the nexthop via the outgoing interface

The NHLFE is located on the transmission interface. That is why NHLFE need not store the outgoing interface.

The Third data structure helps the ingress in deciding which labels to add to the incoming packets. To understand its processing it is important to understand FEC (Forwarding Equivalence Class). The two questions to be tackled are what labels to add to a packet and what type of packet is obtained after the removal of the label.

Packets are labeled according to the FEC they belong to, e.g. all packets destined to the IP address 10.1.1.1 are assigned an FEC of A. In MPLS each FEC is assigned a label while each label refers to an FEC (1:1 mapping). The definition of an FEC may change but the

1:1 mappings remain the same. The data structure that maps FEC to labels is called FEC TO NHLFE (FTN). An FTN table consists of all the FECs that we know how to add labels to. An FTN entry is made up of FEC and NHLFE entry. The FTN process consists of the following steps:

- Decide what FEC a packet belongs to
- Find the FEC in the FTN table
- Forward the packet to the NHLFE that corresponds to the FTN

4.2.3 Implementation of NTP in the Testbed

NTP (Network Time Protocol) is a protocol that is used to synchronize computers in a network. Developed by Davis Mills at the University of Delaware, it is now an Internet standard. NTP uses UTC (Coordinated Universal Time) to synchronize computer clocks up to a fraction of a millisecond.

4.2.3.1 How NTP Operates

The ntpd program exchanges messages with one or more configured servers at designated polling intervals. At startup the program requires several exchanges from the majority of the servers so that signal processing and mitigation algorithms can help to set the clock. The initial poll interval for each server is delayed an interval randomized over a few seconds in order to protect the network from bursts. It can take several minutes before the clock is set.

A TOY (Time of Year) chip is used to maintain the time during periods when the power is off in most of the operating systems and hardware used today. The chip is used to initialize the operating system time at bootup. Once synchronized to an NTP server the operating system corrects the chip from time to time. In the absence of a TOY chip or in case the offset from the server is more than a thousand seconds then the operator has to intervene manually and set the clock himself as ntpd assumes something maybe terribly wrong.

In ordinary conditions, ntpd adjusts the clock in small steps so that the time scale is effectively continuous. Under extreme network congestion, the roundtrip delay jitter might exceed three seconds and the synchronization delay can become very large. Sample offsets exceeding 128 ms are discarded by the ntpd unless the interval during which no sample offset is less than 128 ms exceeds 900 seconds. The first sample following it steps the clock to the indicated time. As a result, once the clock has been set, it very rarely strays more than 128 ms even under extreme network congestion.

4.2.3.2 Deployment of NTP in the Testbed Network

A local NTP server was used to synchronize the senders and receivers. The times on the senders and receivers were set manually initially before running the ntpd program on each of them so that they could synchronize with the NTP server without any errors. The NTP server was running the ntpd program in server mode while the senders and receivers were running the ntpd program in client mode. After it was ensured that the exchanges were taking place between the NTP server and the NTP clients, they were left to synchronize over a period of 24 hours under no additional traffic between the NTP server and the NTP

clients in order to determine accurately delay and jitter for voice traffic up to a fraction of a millisecond. The NTP server and the clients were connected in a manner shown in figure 4.2.

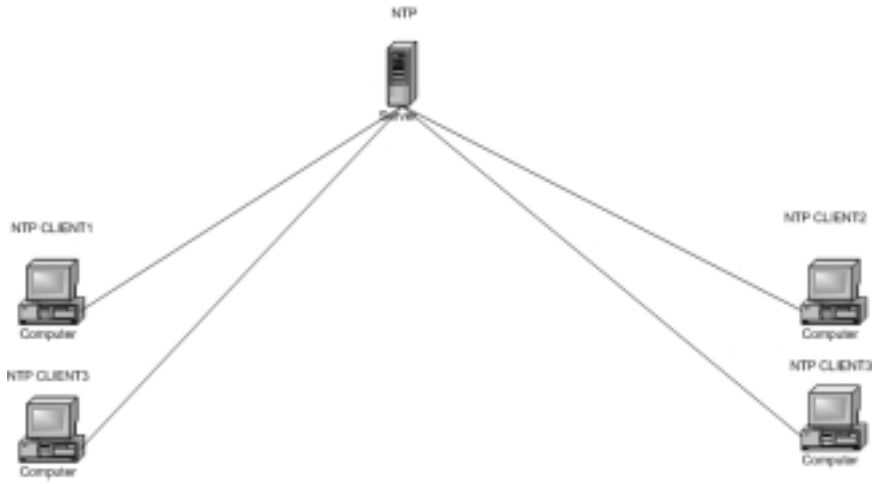


Figure 4. 2 NTP Server setup.

4.3 Workload Selection

4.3.1 Characteristics of Packetized Voice

In packet voice applications, speech is transported as "data" packets, and these packets are generated only when there is actual speech to transport. The elimination of wasted bandwidth during periods of silence will, by itself, reduce the effective bandwidth required for speech transport by approximately one-third.

Both Linear Prediction Coding (LPC) and Pulse Code Modulation/Adaptive Differential Pulse Code Modulation (PCM/ADPCM) coding of voice information are standardized by the ITU in its G-series recommendations. The most popular voice coding standards for telephony and packet voice include the following:

- G.711, which describes the 64 kbps PCM voice coding technique. G.711-encoded voice is already in the correct format for digital voice delivery in the public phone network or through PBXs.
- G.726, which describes ADPCM coding at 40, 32, 24, and 16 kbps. ADPCM voice may also be interchanged between packet voice and public phone or PBX networks, providing the latter has ADPCM capability.
- G.728, which describes code-excited linear-predictive (CELP) voice compression, requiring only 16 kbps of bandwidth. CELP voice coding must be transcoded to a public telephony format for delivery to or through telephone networks.
- G.729, which describes adaptive CELP (ACELP) compression that enables voice to be coded into 8 kbps streams. There are four forms of this standard, and all provide speech quality as good as that of 32 kbps ADPCM.
- G.723.1, which describes a coded representation that can be used for compressing speech or other audio signal component of multimedia services at a very low bit rate as part of the overall H.324 family of standards. This coder has two bit rates associated with it—5.3 and 6.3 kbps. The higher bit rate has greater quality; the lower bit rate gives good quality and provides system designers with additional flexibility.

The voice quality of a compression strategy has been measured by survey—the Mean Opinion Score (MOS) was the first commonly available measurement. On the MOS scale, where zero is poor quality and five is high, the standard PCM has a quality of about 4.4,

G.726 ADPCM is rated at 4.2 for the 32 kbps version. G.728 CELP coding achieves a rating of 4.2, and G.729 a score of 4.2. MOS scores are not standard and the results depend on the particular survey cited, as well as the language and gender mix of the participants.

A more objective measurement has become available and is quickly overtaking MOS scores as the industry quality measurement of choice for coding algorithms. Perceptual Speech Quality Measurement (PSQM), as per ITU standard P.861, also provides for a rating on a scale of zero to five, but here a rating closer to zero is better and five is the worst. Various vendors' test equipment is now capable of providing a PSQM score for a test voice call over a particular packet network.

Packet voice coding improves network economics in two ways; first by reducing the bandwidth consumed by voice traffic, and second by eliminating silent periods. In order to take advantage of these benefits, the underlying transport network must be able to support small-bandwidth traffic streams, and interleave other traffic into silent periods in the voice calls to recover the idle bandwidth that packet voice transport produces. The facilities provided to ensure these capabilities vary depending on the type of network.

4.3.2 Workload Parameters

The voice traffic was exponentially produced at an average rate of 17 packets per second with a payload of 172 bytes considering an ON time of 1.004 seconds and an OFF time of 1.587 seconds with voice transmission rate of 80Kbps during ON time [45]. These values follow the principles given in [46]. According to these principles, when the source is in

the "on" state, fixed-size packets are generated at a constant interval. No packets are transmitted when the source is "off". A realistic reproduction of aggregated VoIP traffic multiple flows with these characteristics were produced simultaneously. E.g. two such flows will be produced exponentially at an average rate of 34 packets per second, three flows at an average of 51 packets per second and so on.

The background traffic was produced at increasing rates of 1000 packets per second. The packet size for background traffic was kept constant at 1KByte. One reason for not modeling background traffic according to Pareto distribution was that it could have lead to unpredictable results, so only voice traffic was modeled using exponential distribution.

4.3.3 Traffic Generation Tools

The Distributed Internet Traffic Generator (D-ITG) is a platform [47], which can produce traffic (network, transport and application layer) and generate stochastic processes for both Inter Departure Time (IDT) and Packet Size (PS) random variables (exponential, uniform, cauchy, normal, pareto …). The capabilities of this traffic generation software were found to be appropriate for the generation of all the required types of traffic according to the experimental requirements which were based on the experiments mentioned in [45].

4.4 Experimental Design

4.4.1 Traffic Patterns

Network traffic is either symmetric or asymmetric in nature i.e. when all the devices in a network are transmitting and receiving data at equal rates then the traffic is said to be symmetric. In asymmetric networks more bandwidth is allocated in one direction than the other.

The voice traffic model used for this study is a mixture of symmetric and asymmetric in nature. The reason for this is that the voice traffic modeled is emulating voice traffic on the Internet taking into consideration all types of coders and silence durations. Even though transmission of voice traffic from all end devices is being done at the same time, at no point between the starting and stopping time is the behavior of voice traffic symmetric in nature as packet generation is exponentially distributed. However the symmetric nature comes from the fact that the bandwidth allocated in either direction is the same.

4.4.2 Comparison of Router Performance across IP and MPLS Delay and Jitter

Performance comparison of IP and MPLS based routers must be conducted under specific workload parameters and factor levels. The two important performance metrics which had to be used for comparison of both routers were delay and jitter.

4.4.3 Comparison of IP and MPLS Router Performance using Drop Rates

The three types of traffic used were UDP based voice traffic, UDP based voice traffic accompanied by TCP based background traffic and UDP based data traffic. Drop rates for each of these three traffic flows were individually determined in both types of network (i.e. IP and MPLS).

4.4.4 Comparison of IP and MPLS Router Performance in Point-to-Point and Point-to-Multipoint Modes

Experiments had to be conducted in point-to-point and point-to-multipoint modes for all the three above mentioned traffic flows to determine performance of both the networks under stressful conditions.

4.4.5 Performance Metrics

The choice of performance metrics was meant to evaluate the routers processing and routing speed under network congestion. Some of the metrics used were as follows:

Router CPU Utilization

This refers to the overall router CPU time that is spent in doing useful work.

Router Memory Utilization

This refers to the amount of memory utilized by the router during the routing process.

End-to-End Delay

It indicates the amount of time a packet takes to traverse the entire network.

Jitter

The interarrival jitter is defined to be the mean deviation (smoothed absolute value) of the difference in packet spacing at the receiver compared to the sender for a pair of packets.

CHAPTER 5

5 MEASUREMENT BASED EVALUATION

5.1 Factor Analysis

The measurement based experiments use congestion levels as the main factor that influences router performance. Since the voice traffic emulated is coder independent and encompasses the behavior of voice traffic on the Internet, the need to find the effects of various coders (like G.711, G.723, G.729, etc) on the routing capabilities of both IP and MPLS based routers is beyond the scope of this research [48].

5.2 Router performance

In order to find how parameters like CPU utilization, memory utilization and number of interrupts will effect the performance of MPLS based software routers as compared to IP based software routers, individual router performance was measured in terms of CPU utilization, number of interrupts per second and the amount of virtual memory that is active on a particular router for the duration of the experiment with varying levels of network congestion.

Figure 5.1 shows the amount of virtual memory that is active on the MPLS and IP based Ingress, Core and Egress routers for point-to-point and point-to-multipoint traffic. Since the MPLS based Ingress router has to add an additional header (shim header) on each

packet it is utilizing more virtual memory. As LSR is only switching packets, its effect on the amount of virtual memory being utilized is evident. MPLS based Egress is doing additional work of popping labels therefore the amount of virtual memory being used is also greater.

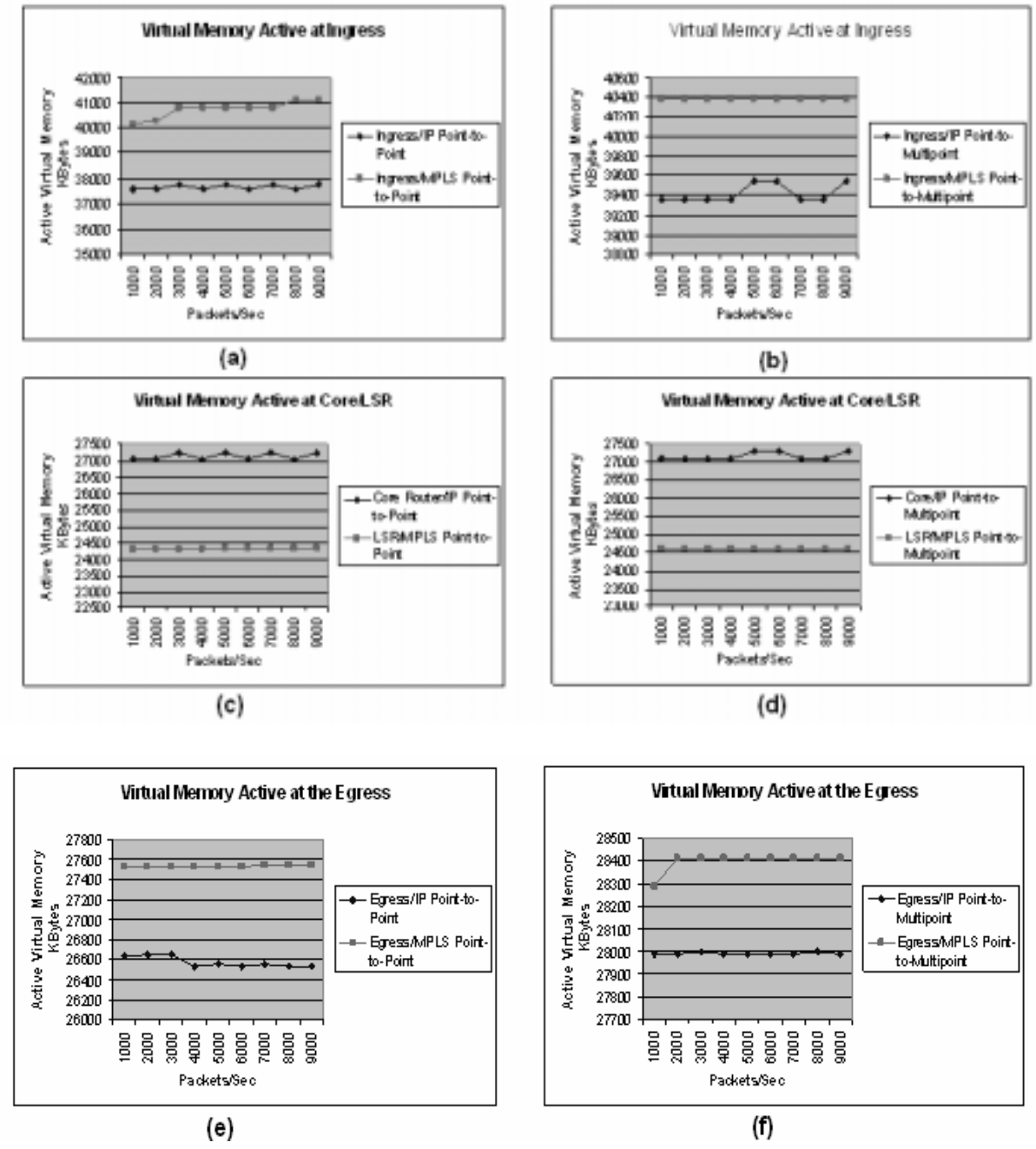


Figure 5.1 Amount of Virtual Memory active at the Ingress, Core and Egress Routers.

Figure 5.2 shows the CPU utilization for voice traffic when accompanied by TCP based background traffic on MPLS and IP based networks for point-to-point and point-to-multipoint traffic.

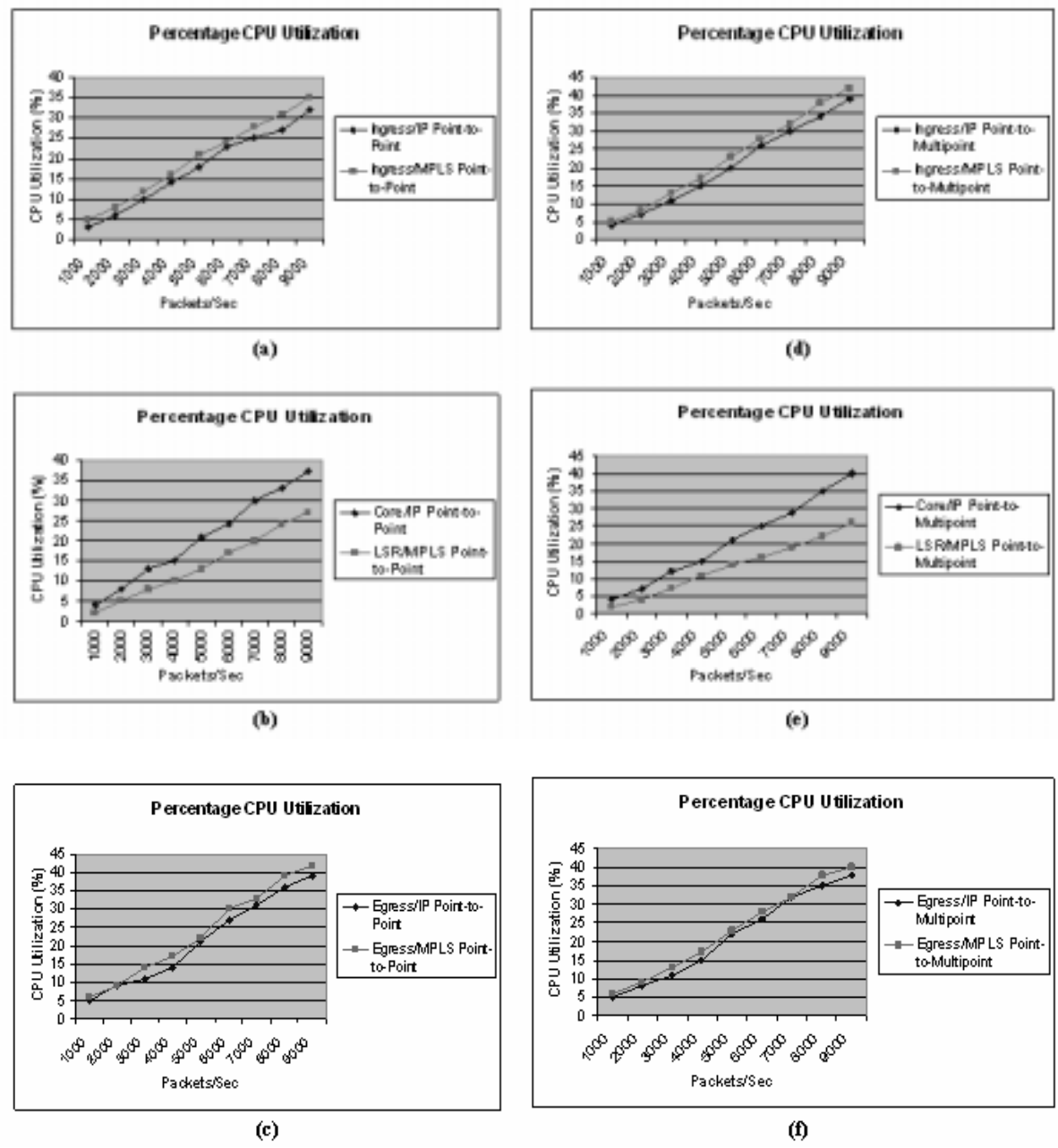


Figure 5.2 Percentage CPU Utilization across Ingress, Core and Egress routers for voice traffic which is accompanied by background traffic.

Since the MPLS based Ingress router does additional work in adding shim header on each packet, its percentage CPU utilization exceeds that of the IP based Ingress router. As LSR is only switching packets, a decrease in CPU utilization is observed at the LSR. At the Egress of the MPLS network additional work is being done in terms of popping labels which explains the increase in percentage CPU utilization as compared to IP based Egress router.

Figure 5.3 shows the number of interrupts per second generated for voice traffic when accompanied by TCP based background traffic on MPLS and IP based networks for point-to-point and point-to-multipoint traffic. Since the MPLS based Ingress router is adding additional header on each packet, the number of interrupts generated per second also increases proportionally. At the LSR, because of the switching of packets only, the number of interrupts generated per second is also distinctly reduced. MPLS based Egress is doing additional work of popping labels thereby causing additional interrupts to be generated per second.

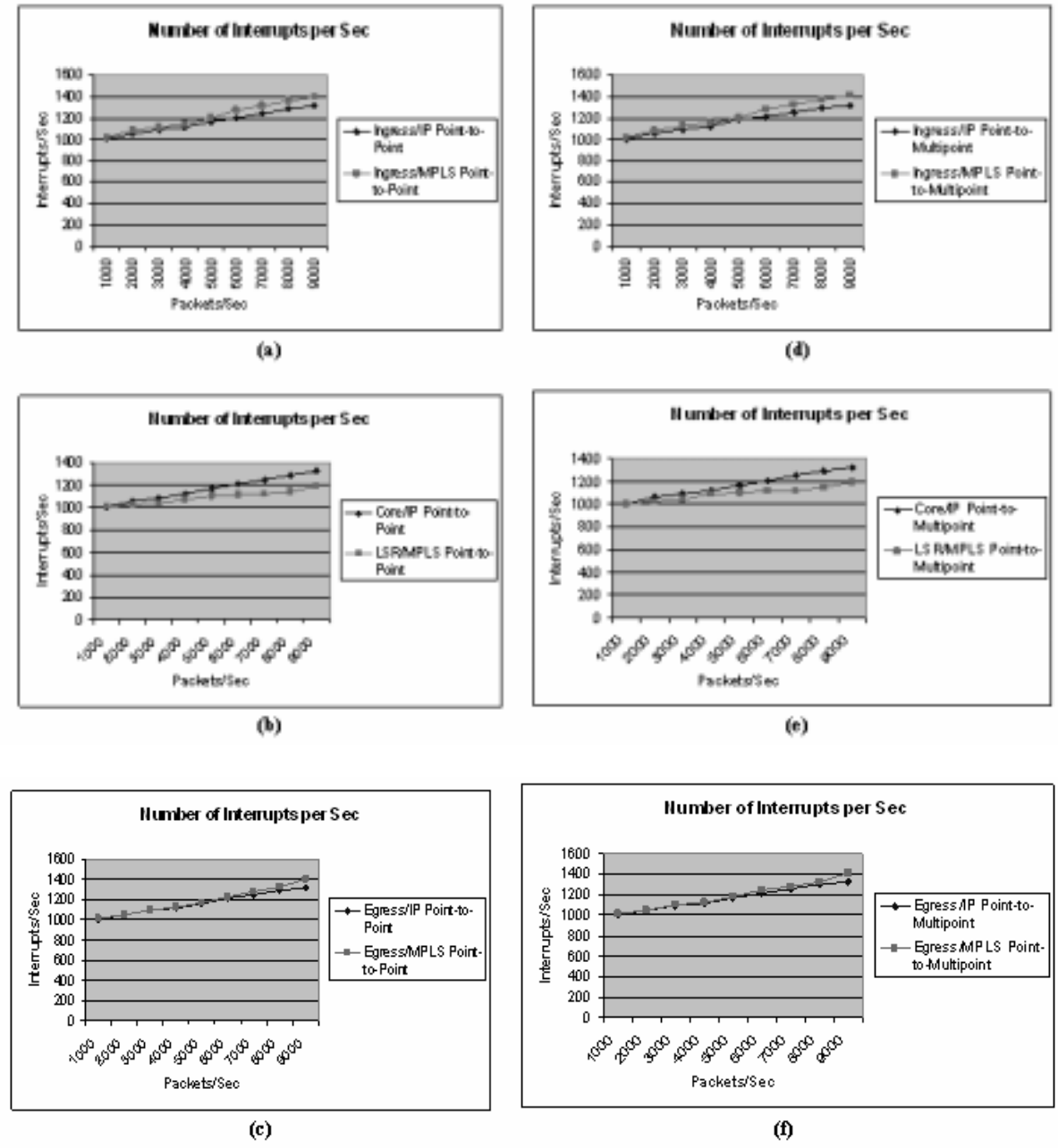


Figure 5.3 Number of Interrupts per Second across Ingress, Core and Egress routers for voice traffic which is accompanied by background traffic.

5.3 Some Limitations of the Traffic Generation Softwares

Before the detailed analysis of the results is presented, it is important to note some limitations of the traffic generation softwares used. A drawback of D-ITG was that it was unable to produce exponentially distributed voice traffic as it only produced one voice coder based traffic at a time. However the payload was adjusted in order to accommodate RTP header when producing exponentially distributed UDP traffic. D-ITG also does not have the capability to produce traffic with ON and OFF times.

5.4 Test, Results and Analysis

As shown in figure 4.1, voice traffic accompanied by TCP based background traffic traversed the network in point-to-point as well as point-to-multipoint fashion. The PC based routers were run at runlevel 3 and the daemons stopped were sshd, xinetd, sendmail, crond and atd. This allowed the processors to do routing more efficiently. Each run spanned a period of one minute. The background traffic was sequentially increased on each run.

Figure 5.4 shows the average end-to-end delay, inter-arrival jitter and drop rate for only voice traffic on MPLS and IP based networks for point-to-point and point-to-multipoint traffic. Rate of voice traffic was increased at 850 packets per second.

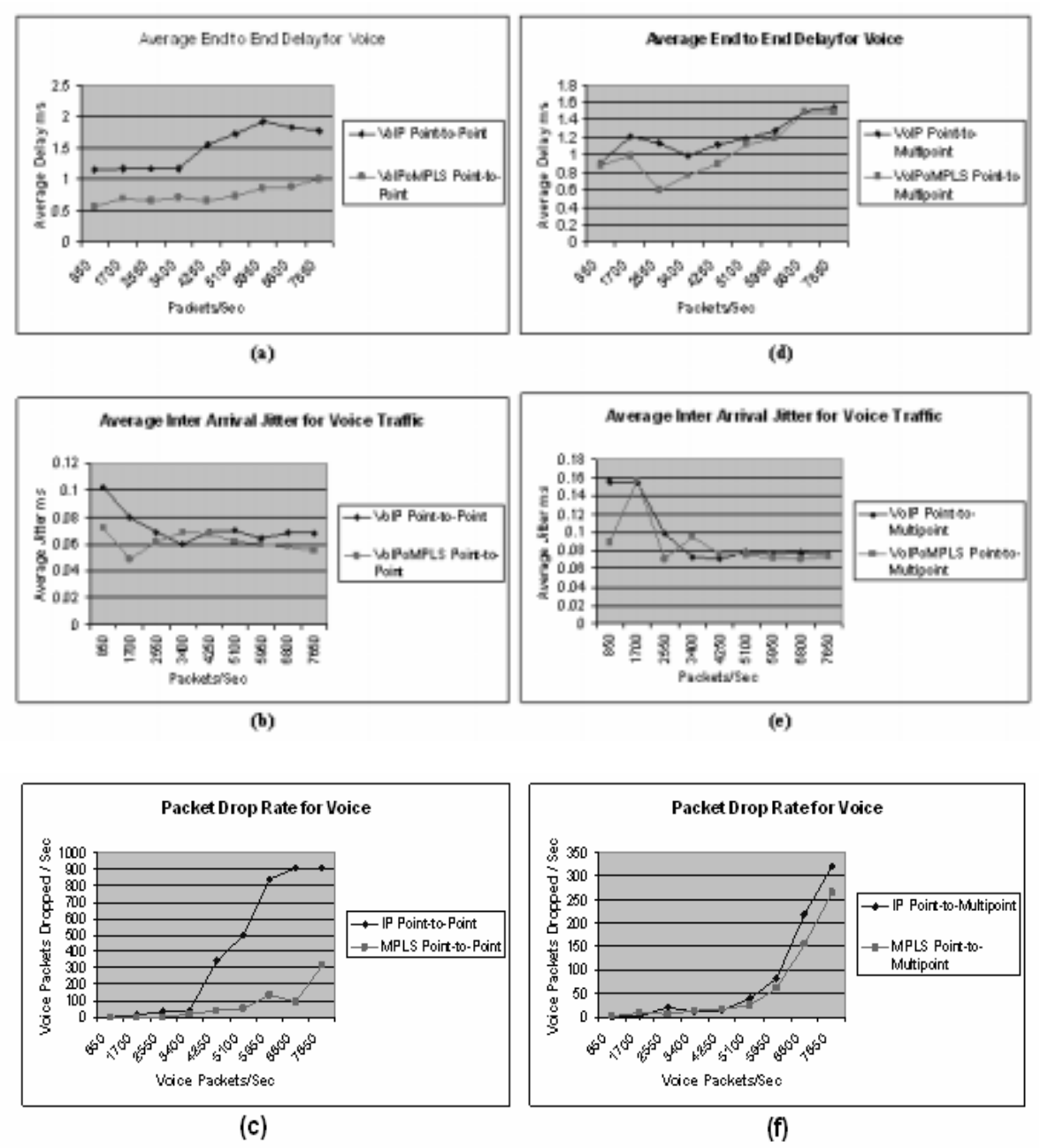


Figure 5.4 Average End-to-End Delay, Inter Arrival Jitter and Drop Rate for only voice traffic.

Figure 5.5 shows the average end-to-end delay for UDP traffic on MPLS and IP based networks for point-to-point and point-to-multipoint traffic. The UDP packet size used was 1 KByte. The rate of packets generated was increased at 1000 packets per second.

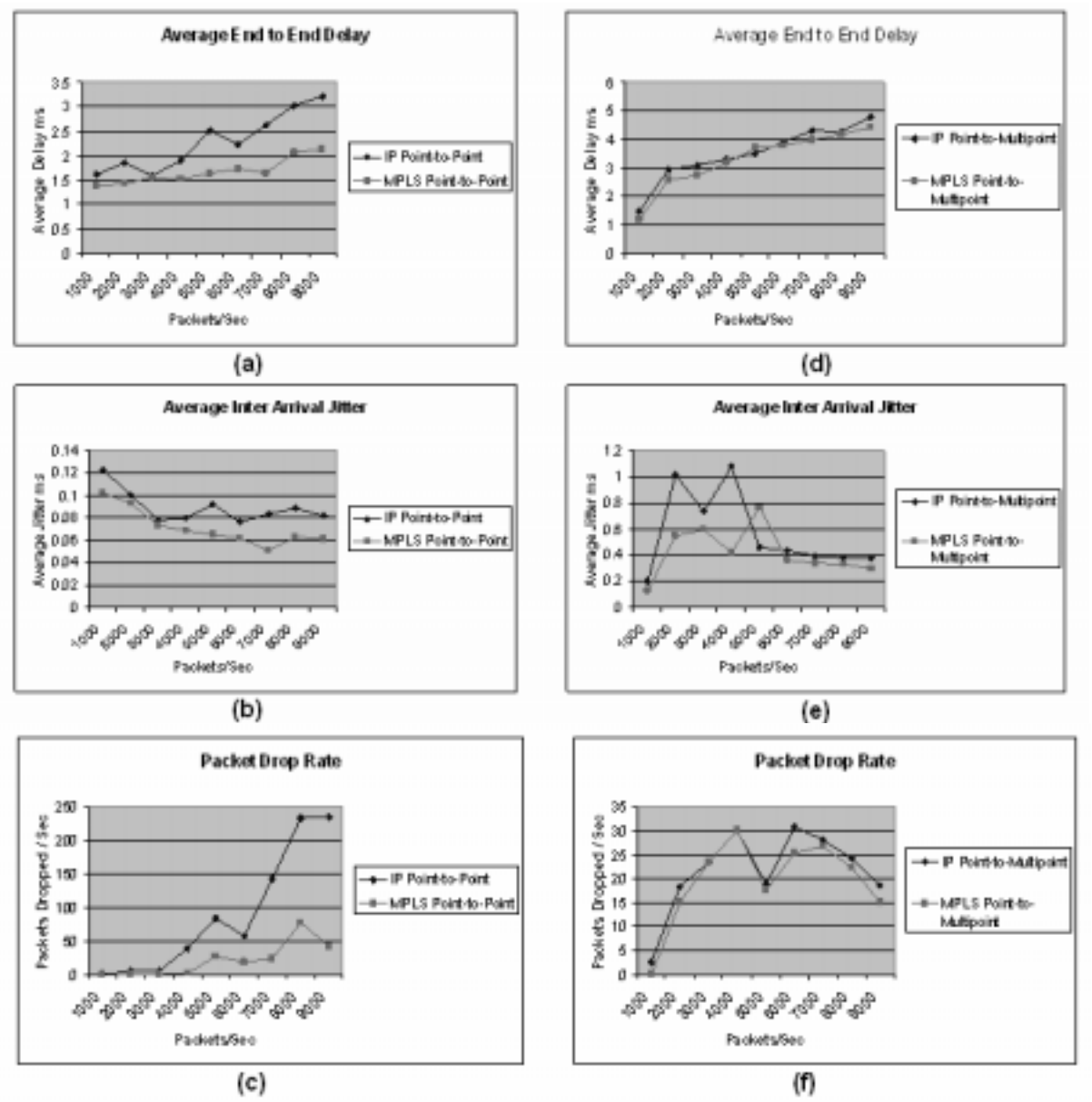


Figure 5.5 Average End-to-End Delay, Inter Arrival Jitter and Drop Rate for only UDP traffic.

Figure 5.6 shows the average end-to-end delay, inter arrival jitter and drop rate for voice traffic when accompanied by TCP based background traffic on MPLS and IP based networks for point-to-point and point-to-multipoint traffic. Rate of voice traffic was kept constant at 850 packets per second and rate of TCP based background traffic was gradually increased by thousand packets per second with each packet of 1 KByte.

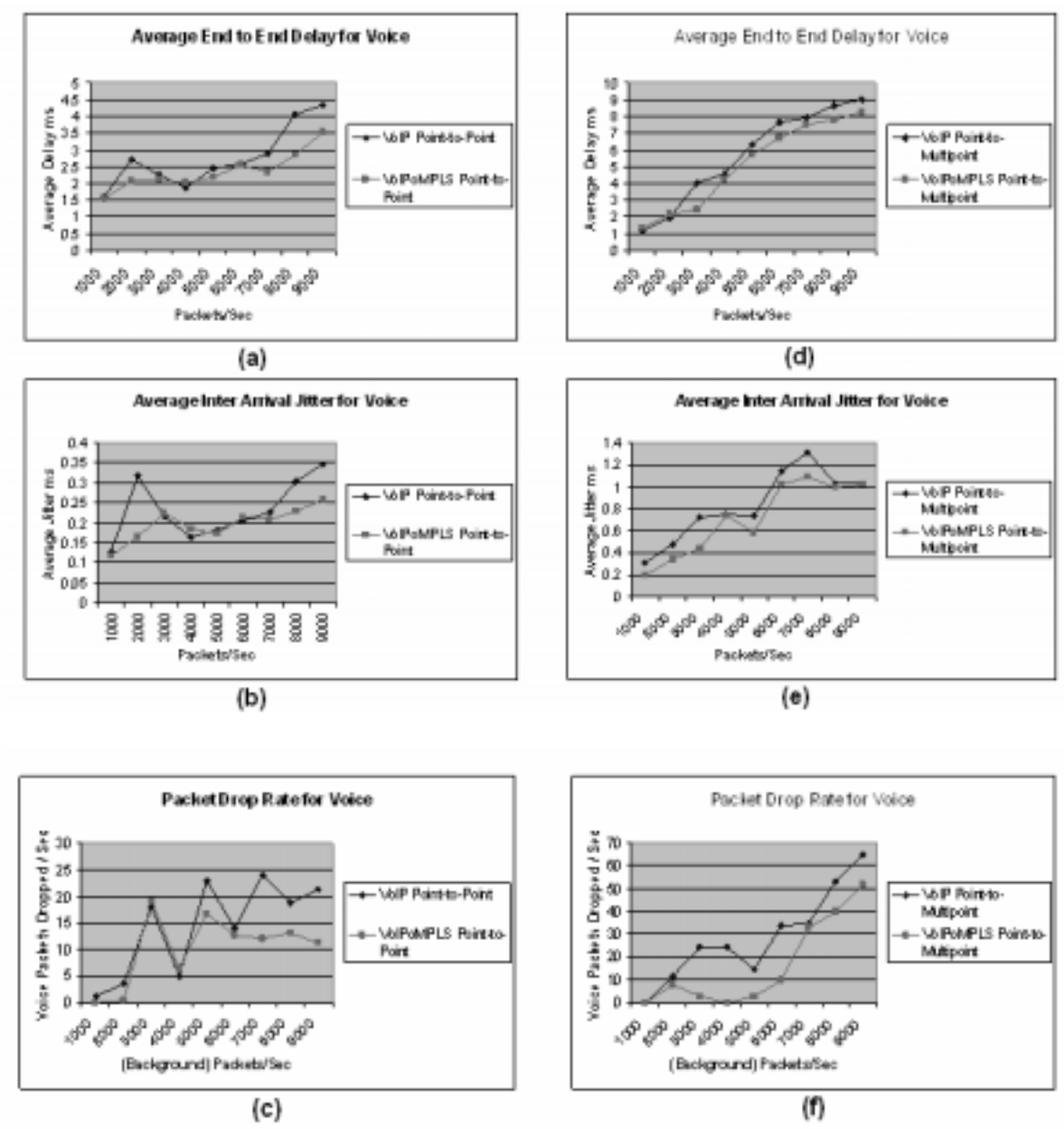


Figure 5.6 Average End-to-End Delay, Inter Arrival Jitter and Drop Rate for voice traffic that is accompanied by background traffic.

Figure 5.7 shows the per packet delay for voice traffic when accompanied by TCP based background traffic on MPLS and IP based networks for point-to-point and point-to-multipoint traffic. Since the MPLS based Ingress router has to add a shim header on each packet, per packet delay across Ingress also increases. As LSR is only switching packets,

per packet delay goes down appreciably. MPLS based Egress is doing additional work of popping labels which is the reason for higher per packet delay.

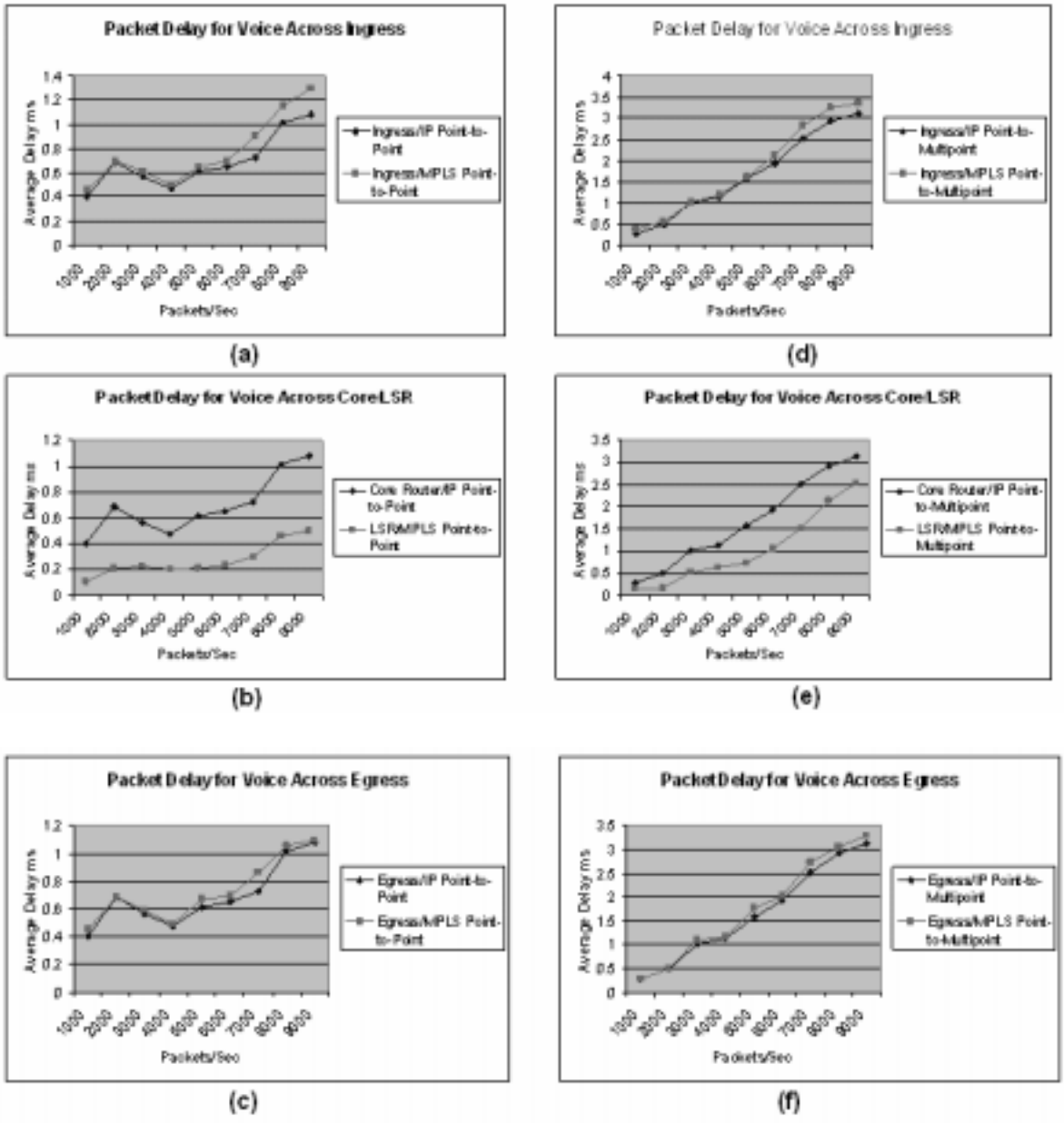


Figure 5.7 Per Packet Delay across Ingress, Core and Egress routers for voice traffic which is accompanied by background traffic.

5.5 Summary of Results

The most striking aspect among all the results obtained is the extent of advantage gained in the core of the MPLS network through packet switching. The average end-to-end delay, average inter arrival jitter along with the packet drop rate have been reduced significantly in the MPLS network as compared to IP network even though additional processing overhead is observed at the Ingress and the Egress routers in the MPLS network. The greater the number of LSRs in the core of the network, the greater will be the benefit gained out of it in terms of the above mentioned parameters.

Voice quality drops considerably when the end-to-end delay exceeds 150 ms [49], which is not observed in both setups i.e. for MPLS and IP. Based only on this fact it would seem that the improvement in the quality of voice provided by MPLS compared to IP is negligible. However when the delay increases with the increase in traffic and increase in the number of core routers in both kinds of networks, the comparison between the overall delays of both networks will definitely become much more significant. There will come a time when the delay for voice traffic in IP network exceeds 150 ms whereas the delay in MPLS network remains below 150 ms. An additional benefit provided by the MPLS network is that the complexity of the network moves to the edges which reduces the number of network elements to manage. The reason is that the criteria on which LSPs are to be setup and incoming and outgoing traffic is to be mapped are defined on Ingress and Egress routers only and the LSRs only need instructions on how to switch labels and route accordingly. Also Layer 2 independence provided by MPLS makes it more scalable and flexible and traffic engineering allows better control over different flows of traffic.

CHAPTER 6

6 CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

The layer 2 technology used in experimentation was Ethernet that is why when we say VoIPoMPLS we actually mean VoIPoMPLSoEthernet. In this research we have compared the performance of VoIPoEthernet with VoIPoMPLSoEthernet with respect to delay, jitter and drop rate. The main purpose of the research was to find the advantage gained in terms of time when it comes to both one way and two way voice communication taking place over a backbone network when employing simple IP based forwarding or MPLS based routing/switching.

Theoretical intuition or hypothesis suggests that the benefit gained out of MPLS based switching in the core of the backbone through the use of LSRs will surely outperform IP based forwarding. This was the reason why a number of LSRs were introduced in the testbed backbone network. In order to obtain accurate delay and jitter value for both technologies, an NTP server was introduced in the topology which kept all the senders and receivers in sync.

Even though PC based routers are never employed in actual enterprise networks because of their non-specialized hardware and unpredictable behavior they can however be used to test new technologies and scenarios.

6.2 Future Research

The future work has to focus on the vendor based implementation of MPLS in their routers and switches to get a true picture of the difference in performance of the two technologies when it comes to supporting real time traffic like voice.

New technologies like Generalized MultiProtocol Label Switching and Layer 2 Tunneling Protocol are also grabbing the attention of the networking community because of the advanced features they support and are also in the hunt to capture a large part of the networking business. Research can be done on such upcoming technologies and a detailed analysis of their performance can go a long way in helping the networking community.

BIBLIOGRAPHY

[1] Uyless Black. “Voice Over IP.” Prentice Hall, Upper Saddle River, NJ, 2002.

[2] Li, T. “MPLS and the evolving Internet architecture”. Communications Magazine, IEEE, Volume: 37 Issue: 12, Dec. 1999Page(s): 38 –41.

[3] **“**Protection and Restoration in MPLS Networks”, DATA Connection, http://www.dataconnection.com/download/mplsprotwp2.pdf, February 2002**.**

[4] “Multiprotocol Label Switching”, Search Networking.com, http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci214350,00.htm, February 2002

[5] Yassir Obeid Mohammed, “Quality of Service Routing”, M.S. Thesis, Computer Engineering Department, KFUPM, 2001

[6] F. Le Faucheur et al., "MPLS Support of Differentiated Services", IETF Draft, draft-ietf-mpls-diff-ext-07.txt, August 2000.

[7] IETF Working Groups, http://ccl.cnu.ac.kr/IETF/IntServ/intserv.htm

[8] S. Blake et al., “An Architecture for Differentiated Services,” RFC 2475, 1998

[9] Differentiated Services on the Internet, http://kabru.eecs.uwich.edu/qos_network/diffserv/DiffServ.html

[10] Script: ATM Audio Primer, http://www.nwfusion.com/primers/atm/atmscript.html

[11] Rouhana, N.; Horlait, E.; “Differentiated services and integrated services use of MPLS”, Computers and Communications, 2000. Proceedings. ISCC 2000. Fifth IEEE Symposium on, 3-6 July 2000 Page(s): 194 –199.

[12] F. Tommasi, S. Molendini, A. Tricco: "Mapping of IntServ/RSVP reservations into MPLS domains", in Proceedings of International Conference on Software, Telecommunications and Computer Networks IEEE SOFTCOM2002, 8-11 October 2002, Split, Croatia.

[13] Kankkunen, A.; "MPLS and next generation access networks", Universal Multiservice Networks, 2000. ECUMN 2000. 1st European Conference on, 2-4 Oct. 2000 Page(s): 5 –16.

[14] Moh, M.; Wei, B.; Zhu, J.H.; "Supporting differentiated services with per-class traffic engineering in MPLS", Computer Communications and Networks, 2001. Proceedings. Tenth International Conference on, 15-17 Oct. 2001 Page(s): 354 –360

[15] F. Le Faucheur, et al., "MPLS Support of Differentiated Services," draft-ietf-mpls-diff-ext-09&t, April 2001

[16] Rosen E., Viswanathan A., Callon R., "Multiprotocol Label Switching Architecture", Internet Draft, <draft-ietf-mplsarch -01.txt>, March 1999

[17] Moh, W.M.; Liming Xiang; Xiang Zhao; Eun Park; "Differentiated-service-based inter-domain multicast routing: enhancement of MBGP", Computer Communications and Networks, 2000. Proceedings. Ninth International Conference on, 16-18 Oct. 2000 Page(s): 290 –297

[18] Tae-Won Lee, 'Implementation of a MPLS Router Supporting DiffServ for QoS and High Speed Switching' IEEE Communications – May 2002

[19] Gang Yuan; Wendong Wang; Yu Lin; Shiduan Cheng; "A QoS management architecture for diffserv-aware MPLS-based IP network", Communication

Technology Proceedings, 2003. ICCT 2003. International Conference on, Volume: 2, April 9 - 11, 2003 Page(s): 1603 –1607

[20] K. Chan, et al. “COPS Usage for Policy Provisioning (COPS-PR),” RFC 3084, March 2001.

[21] Trimintzios, P.; Andrikopoulos, I.; Pavlou, G.; Flegkas, P.; Griffin, D.; Georgatsos, P.; Goderis, D.; T'Joens, Y.; Georgiadis, L.; Jacquenet, C.; Egan, R.; “A management and control architecture for providing IP differentiated services in MPLS-based networks,” Communications Magazine, IEEE, Volume: 39 Issue: 5, May 2001 Page(s): 80 –88

[22] Shahsavari, M.M.; Al-Tunsi, A.A.; “MPLS performance modeling using traffic engineering to improve QoS routing on IP networks,” SoutheastCon, 2002. Proceedings IEEE, 5-7 April 2002 Page(s): 152 –157

[23] Autenrieth, A.; Kirstadter, A.; “RD-QoS - the integrated provisioning of resilience and QoS in MPLS-based networks,” Communications, 2002. ICC 2002. IEEE International Conference on, Volume: 2, 28 April-2 May 2002 Page(s): 1174 -1178 vol.2

[24] Fineberg, V.; Cheng Chen; XiPeng Xiao; “An end-to-end QoS architecture with the MPLS-based core,” IP Operations and Management, 2002 IEEE Workshop on, 2002 Page(s): 26 –30

[25] Marzo, J.L.; Calle, E.; Scoglio, C.; Anjali, T.; “Adding QoS protection in order to enhance MPLS QoS routing,” Communications, 2003. ICC '03. IEEE International Conference on, Volume: 3, 11-15 May 2003 Page(s): 1973 –1977

[26] Karol Kowalik and Martin Collier, “QoS routing as a tool of MPLS Traffic Engineering,” First Joint IEI/IEE Symposium on Telecommunications Systems Research, November 2001, Dublin, Ireland

[27] Sangsik Yoon; Hyunseok Lee; Deokjai Choi; Youngcheol Kim; Gueesang Lee; Lee, M.; “An efficient recovery mechanism for MPLS-based protection LSP,” ATM (ICATM 2001) and High Speed Intelligent Internet Symposium, 2001. Joint 4th IEEE International Conference on, 22-25 April 2001 Page(s): 75 –79

[28] Zhiqun Zhang; Xu Shao; Wei Ding; “MPLS ATCC: an active traffic and congestion control mechanism in MPLS,” Info-tech and Info-net, 2001. Proceedings. ICII 2001 - Beijing. 2001 International Conferences on, Volume: 2, 29 Oct.-1 Nov. 2001 Page(s): 222 -227 vol.2

[29] C.Huang, V. Sharma, K. Owens and S. Makam, "Building Reliable MPLS Networks Using a Path Protection Mechanism," IEEE Communications Magazine, March 2002.

[30] Florian-Daniel Otel, "On fast computing bypass tunnel routes in MPLS-based local restoration," HSNMC 2002, Jeju, Korea.

[31] Agarwal, A.; Deshmukh, R.; “Ingress failure recovery mechanisms in MPLS network,” MILCOM 2002. Proceedings, Volume: 2 , Oct. 7-10, 2002 Page(s): 1150 –1153

[32] Zhang, C. and Guy C.G. “TE-SIP Server Design for a SIP-over-MPLS based network,” International Conference on Communications Technology (ICCT 2003), Beijing, China, April 9-11, to appear.

[33] S. Dragos, R. Dragos and M. Collier, "Bandwidth Management in MPLS Networks," Dublin City University

[34] R. Bartoš and A. Gandhi, ``Dynamic issues in MPLS service restoration." Proc. of the Fourteenth IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS), Cambridge, MA, (S. G. Akl and T. Gonzalez, eds.), pp. 618-623, November 2002

[35] Franco Tommasi, Simone Molendini, Andrea Tricco: "Improvement of performance in MPLS domains by using caching and aggregation of CR-LSP," Net-Con 2002: 267-272

[36] Aukia, P.; Kodialam, M.; Koppol, P.V.N.; Lakshman, T.V.; Sarin, H.; Suter, B.; "RATES: a server for MPLS traffic engineering," Network, IEEE, Volume: 14 Issue: 2, March-April 2000 Page(s): 34 –41

[37] Brunner, M.; Quittek, J.; "MPLS management using policies," Integrated Network Management Proceedings, 2001 IEEE/IFIP International Symposium on, 14-18 May 2001 Page(s): 515 –528

[38] Metz, C.; "Layer 2 over IP/MPLS," Internet Computing, IEEE, Volume: 5 Issue: 4, July-Aug. 2001 Page(s): 77 –82

[39] Ikuo Nakagawa, Hiroshi Esaki, Kenichi Nagami: ``A Design of a Next Generation IX using MPLS technology", SAINT2002, Nara, Jan., 2002

[40] Choi, T.S.; Yoon, S.H.; Chung, H.S.; Kim, C.H.; Park, J.S.; Lee, B.J.; Jeong, T.S.; "Wise: traffic engineering server for a large-scale MPLS-based IP network," Network Operations and Management Symposium, 2002. NOMS 2002. 2002 IEEE/IFIP, 15-19 April 2002 Page(s): 251 –264

[41] Radu Dragos, Sanda Dragos and Martin Collier, "Design and implementation of an MPLS based load balancing architecture for Web switching," published in Proceedings of 15th ITC Specialist Seminar, Wurzburg, Germany, 22nd-24th July, 2002

[42] Taesang Choi; Hyungseok Chung; Changhoon Kim; Taesoo Jeong; "Design and implementation of an information model for integrated configuration and performance management of MPLS-TE/VPN/QOS," Integrated Network Management, 2003. IFIP/IEEE Eighth International Symposium on, March 24-28, 2003 Page(s): 143 -146

[43] Wright, D.; "Voice over MPLS compared to voice over other packet transport technologies," Communications Magazine, IEEE , Volume: 40 , Issue: 11 , Nov. 2002 Pages:124 - 132

[44] MPLS for Linux, http://sourceforge.net/projects/mpls-linux/

[45] Christos Bouras, Dimitrios Primpas, Afrodite Sevasti1, Andreas Varnavas, "Enhancing the DiffServ Architecture of a Simulation Environment", Proceedings Sixth IEEE International Workshop on Distributed Simulation and Real-Time Applications, Oct.11-13 2002

[46] ITU-T, P.59, 'Artificial conversational speech', (03/93)

[47] Distributed Internet Traffic Generator, http://www.grid.unina.it/software/ITG/

[48] Raj Jain, "The Art of Computer Systems Performance Analysis", Wiley, April 1991.

[49] Understanding Delay in Packet Voice Networks, http://www.cisco.com/warp/public/788/voip/delay-details.html

[50] MPLS Forum, "Voice over MPLS — Bearer Transport Implementation Agreement," 2001

[51] ITU-T Rec. I.366.2, "AAL Type 2 Service Specific Convergence Sublayer for Trunking," 1998

[52] Frame Relay Forum, "Voice over Frame Relay Implementation Agreement," FRF.11, 1997.

[53] D. J. Wright, Voice over Packet Networks, Wiley, 2001.

[54] E. Rosen et al., RFC3032, MPLS Label Stack Encoding, 2001.

[55] B. Davie et al., "MPLS Using LDP and ATM VC Switching," RFC 3035, 2001.

[56] ATM Forum, "ATM Trunking Using AAL2 for Narrowband Services," 1999.

[57] S. Casner and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links," RFC 2508, 1999.

[58] R. Braden, Ed., "Resource Reservation Protocol (RSVP) — Functional Specification," RFC 2205, 1997.

[59] ATM Forum, "PNNI: Private Network to Network Interface," af-pnni-0055.000, v1.0, 1996.

[60] B. Jamoussi et al., "Constraint-Based LSP Setup Using LDP," RFC 3212, 2002.

[61] D. Awduche et al., "RSVP-TE: Extensions to RSVP for LSP Tunnels," RFC 3209, 2001.

[62] D. Awduche et al., "Requirements for Traffic Engineering over MPLS," RFC 2702, 1999.

[63] Daniel Collins, “Carrier Grade Voice over IP”, Published by R.R. Donnelly & Sons Company, 2001, ISBN: 0-07-136326-2

[64] Uyless Black, “MPLS and Label Switching Networks”, Published by Prentice Hall PTR, 2001, ISBN: 0-13-015823-2

[65] J. Postel, “User Datagram Protocol”, IETF RFC 768, http://www.ietf.org/rfc/rfc0768.txt?number=768, 28. August 1980

[66] “Connectionless Transport: UDP”, Computer Networks Research Group, http://www-net.cs.umass.edu/kurose/transport/UDP.html, February 2002

[67] RTP/RTCP, “VoIP, The Basic”, http://www.sidkhosla.com/papers/voip.doc, March 2002

[68] “Algorithm cuts VoIP bandwidth requirement”, EETimes, http://www.eetimes.com/story/OEG20020108S0054, March 2002

[69] “QoS”,searchNetworking.com, http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci213826,00.html, March 2002

[70] Helge Gundersen & Frode Trydal, “QoS for real-time IP traffic”, Agder University College, http://siving.hia.no/ikt01/ikt6400/ftryda95/Report.doc, March 2002

[71] “Protection and Restoration in MPLS Networks”, DATA Connection, http://www.dataconnection.com/download/mplsprotwp2.pdf, February 2002

[72] “Label-Switching Technique Helps Transmit Voice over IP Networks”, Ben Miller, http://www.integralaccess.com/pdf/vompls.pdf, March 2002

[73] "Voice Quality (VQ) in Converging Telephony and Internet Protocol (IP) Networks", International Engineering Consortium, http://www.iec.org/online/tutorials/voice_qual/index.html, May 2002

[74] "MPLS – Technology and Application", The School of Engineering Science, http://www.ensc.sfu.ca/~ljilja/cnl/presentations/william/mpls/sld023.htm, April 2002

[75] Stephen Vogelsang, "Fulfilling The Promise of MPLS: Ethernet Private Line Services Emerge as a First Killer App", Converge! Network Digest, http://www.convergedigest.com/Bandwidth/archive/010806GUESTstephenvogelsgang1.htm, April 2002

[76] Wroclawski, J., Specification of the Controlled-Load Network Element Service, RFC 2211, 1997.

[77] Shenker S., Partridge C., Guerin R., "Specification of Guaranteed Quality of Service", RFC2212, September 1997.

[78] The ATM forum, http://www.atmforum.com/.

[79] H. Bruyninckx, Real Time and embedded HOWTO, http://people.mech.kuleuven.ac.be/ bruyninc/rthowto/.

[80] Inc Cisco Systems, Internetworking Technology Handbook, http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito doc/index.htm, 2002.

[81] A. Farrel and B. Miller, Surviving failures in MPLS networks. Technical report, Data Connection, February 2001.

[82] IEEE. 802.3-2002 Information Technology - Telecommunication & Information Exchange Between Systems - LAN/MAN - Specific Requirements

- Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications 2002, 2002.

[83] B. Jamoussi, L. Andersson, R. Callon, R. Dantu, L. Wu, P. Doolan, T. Worster, N. Feldman, A. Fredette, M. Girish, E. Gray, J. Heinanen, T. Kilty, and A. Malis. IETF RFC 3212: Constraint-Based LSP Setup using LDP, January 2002.

[84] J. Moy. IETF RFC 2328: OSPF version 2, April 1998.

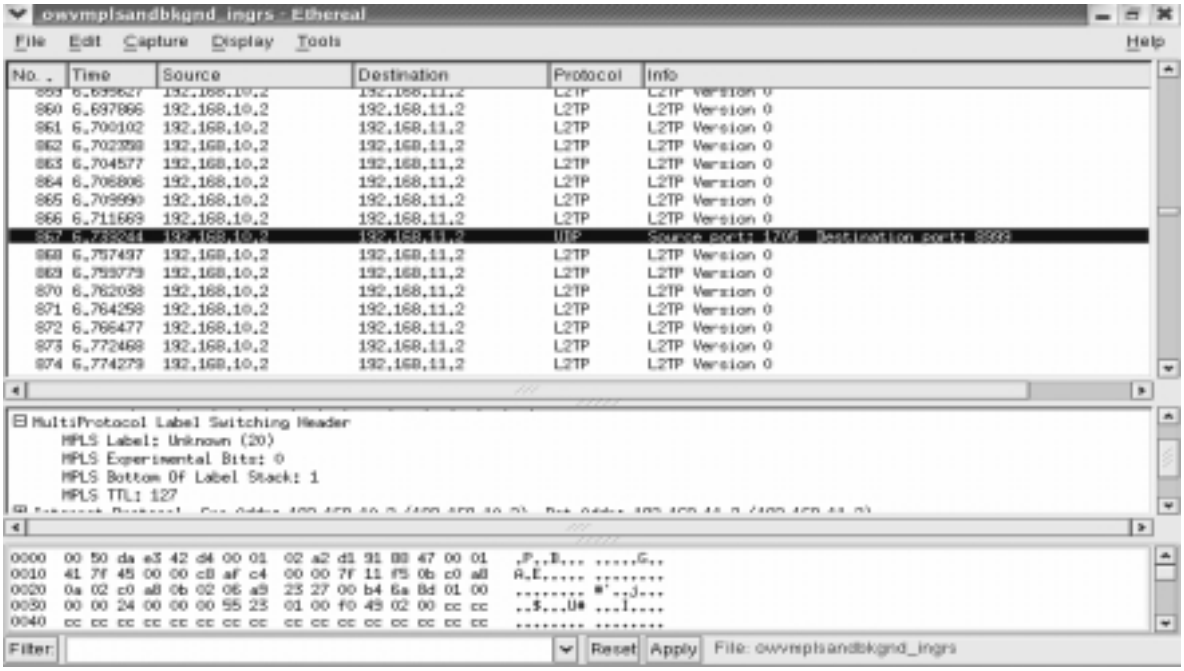
[85] Linux manpages | Netlink.

[86] Y. Rekhter and T. Li. IETF RFC 1771: A Border Gateway Protocol 4 (BGP-4), March 1995.

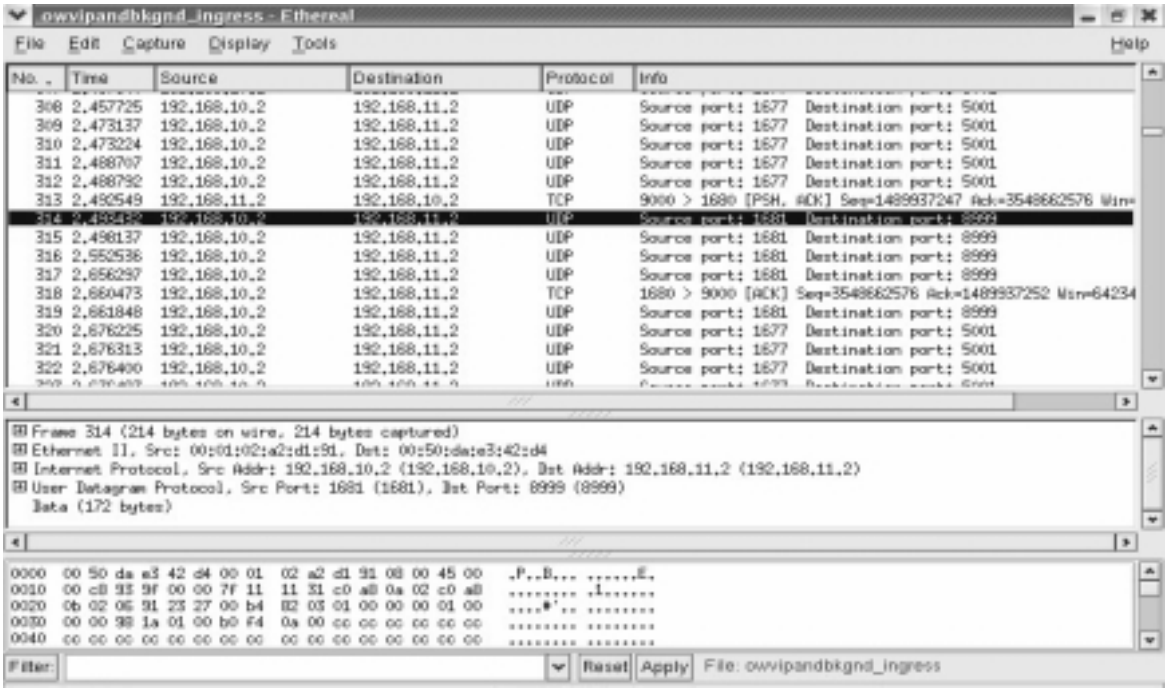
[87] E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li, and A. Conta. IETF RFC 3032: MPLS Label Stack Encoding, January 2001.

APPENDIX A

Appendix A.1: Network trace of voice traffic at the MPLS interface of Ingress



Appendix A.2: Network trace of voice traffic at the IP interface of Egress



VITAE

- **Quadri, Itrat Rasool**
- Born in Pakistan.
- Completed Bachelor of Science (B.S) degree in Computer Engineering from Sir Syed University of Engineering & Technology, Karachi, Pakistan in December 2000.
- Joined Computer Engineering Department, King Fahd University of Petroleum & Minerals (KFUPM) as a Research Assistant in September 2002.
- Completed MS in Computer Engineering from KFUPM, Dhahran, Saudi Arabia in October 2004.
- Email: iquadri@ccse.kfupm.edu.sa