

# Bandwidth Management in MPLS Networks

Sanda Dragos & Radu Dragos  
Supervised by **Dr. Martin Collier**  
School of Electronic Engineering - DCU  
Broadband Switching and Systems Laboratory  
e-mail: {dragoss, dragosr, collierm}@eeng.dcu.ie

November 16, 2001

## Abstract

Two major limitations in legacy IP networking technologies are the impossibility of predicting the bandwidth available on a link at any given time and the difficulty of controlling the allocation of shared bandwidth. Addressing these problems requires a change in the management of networking resources.

Multi Protocol Label Switching (MPLS) is the industry-standard approach to switching and forwarding for next-generation routed networks. The MPLS technology combines the simplified forwarding characteristics of link-layer switching with the control and scalability of network-layer routing.

MPLS reduces the complexity of forwarding using encapsulated fixed-length labels for making high speed forwarding decisions, thus eliminating conventional network-layer header processing. Routers which support MPLS are referred to as Label Switching Routers (LSRs).

Linux is a free, open-source, POSIX compliant, UNIX clone operating system. Its true preemptive multitasking, multiuser support, memory protection and symmetric multiprocessing support characteristics together with its networking, graphical user interface, speed and stability make Linux a preferred tool for research and development.

This project addresses Traffic Engineering (TE) in MPLS networks. The intention is to deploy an improved protocol in a live network. The first step is the implementation of a label switching router in Linux.

**Keywords:** Bandwidth Management, traffic engineering, MPLS, IP, ATM, Linux

## 1 Brief status of communication networks

The current major network technologies in the world are the telephone network and the Internet. Each of them has strengths and weakness. The telephone network offers end-to-end quality-of-service guarantees, while the Internet is more flexible and lower cost.

The telephone network is considered to be the world's largest network since it interconnects more than a billion end-points. Being designed for a single application, to transport analogue and/or digitized voice, the telephone network guarantees voice calls to receive sufficient bandwidth and to have low latency. This is achieved at the loss of efficiency because even when a caller is not speaking, the network continues to reserve resources for that call, while those resources could be used by other callers. That is the reason why the telephone network is more expensive to operate than the Internet, but unlike the Internet it can offer an end-to-end guarantee of quality-of-service.

Started as a research project in the late 1960s, the Internet is a network of networks which nowadays connects tens of millions of computers around the world. The Internet has been very successful because of its scalability, decentralized control and store-and-forward architecture, but these very features make it harder to introduce billing, reliability and quality-of-service.

In recent years, the datacom and telecom communities have tried to find a solution to the problem of how best to build a multi-purpose network. One approach to this problem is to design a new network that has the flexibility and low cost of the Internet, yet offers the end-to-end quality-of-service guarantees of the telephone network. This is the goal of Asynchronous Transfer Mode (ATM) networks. Another approach is to add a quality-of-service mechanism to the Internet.

All those efforts are intended to provide better services for users and to improve the management of available bandwidth.

## 2 Bandwidth management

Bandwidth Management systems address the issue of bandwidth allocation to applications, users, and organizations sharing the same networking resources.

A good management scheme should[1]:

- enable full control of network traffic;
- ensure the availability of bandwidth for critical applications;
- increase network efficiency and reduce traffic congestion.

The other aspect of an Bandwidth Management system is concerning the possibility of gathering detailed information about the dynamics of the network. Binding network-use statistics to databases allows Internet Service Providers (ISPs) to provide differentiated classes of services to users and ensure end-to-end quality of services (QoS).

Current limitations of IP protocols can be overcome by deploying new technologies for next-generation routed networks which can guarantee QoS and have natural support for popular Internet services like virtual private networks(VPN).

The process of optimizing the performance of operational networks is called traffic engineering (TE). It has the ability to analyze current traffic load and dynamically make the necessary changes to achieve the performance needed for a certain bandwidth requirement. MPLS is one of the proposed technologies

useful for traffic engineering in the specific aspects of measurement and control of Internet traffic.

## 3 Brief history and status of MPLS

The idea of MPLS originate from two sources: a faster routing mechanism for IP and providing ATM switches with the control and scalability of a *layer3* router.

In the mid 1990s, few proposals emerged from three major networking companies [2]:

- **Tag Switching**, from Cisco Systems;
- **SITA (Switching IP Through ATM)**, from Telecom Finland;
- **ARIS (Aggregate Route-based IP Switching)**, from IBM.

To develop a standard approach for switching IP the IETF MPLS working group was established in early 1997. In addition, many Internet-drafts related with the development of MPLS were posted by individual contributors or organizations.

1999 saw the first MPLS VPN and traffic engineering deployments. However, the first Internet-drafts became RFCs only in January 2001 while there are many drafts in progress.

The status of MPLS standard as proposed in January 2001 is specified in the following RFCs:

- RFC 3031 - *Multiprotocol Label Switching Architecture* [3]
- RFC 3032 - *MPLS Label Stack Encoding* [4]
- RFC 3035 - *MPLS using LDP and ATM VC Switching* [5]
- RFC 3036 - *LDP Specification* [6]
- RFC 3037 - *LDP Applicability* [7]

## 4 Label switching paradigm and MPLS

Label switching paradigm consists in using a short, fixed-length label to perform switching decisions.

Unlike *longest prefix match* lookup algorithms used by standard IP routing protocols, label switching is based on an exact match and therefore is much faster.

MPLS is a "multiprotocol" which uses label switching technology. The routers supporting MPLS are referred to as Label Switching Routers (LSRs). Any other router or switch connected to a LSR (ATM switch, IP router) is referred to as non-LSR. An edge router is an LSR connected to a non-LSR. The router by which a packet enters the MPLS cloud is called the ingress LSR, and the one by which it leaves the MPLS cloud is called the egress LSR. Label Switching Path (LSP) is the route within the cloud, followed by a packet based on his label.

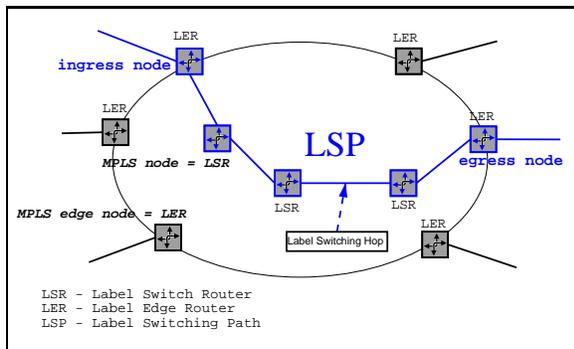


Figure 1: Encapsulation of MPLS packets

Labels are small identifiers inserted by the ingress LSR, and removed by the egress LSR. They are used in forwarding decisions representing the destination of the packet and also the specific path for reaching this destination and attributes of a specific service class.

This section contains a brief description of the advantage of label switching. A more complete description can be found in [8].

- **speed and delay** -label switching is an efficient solution to the problem of large traffic loads in the Internet by using a faster routing table lookup mechanism.
- **jitter** -label switching operations result in less delay and less jitter in sending user traffic through the network.
- **scalability** -label switching offers solutions to the

rapid growth of networks by allowing a large number of IP addresses to be associated with one or a few labels.

- **simplicity** -forwarding decisions are based on a short, fixed-length label.
- **resource consumption** -the control mechanisms to establish label switching paths (LSP) do not consume a lot of resources.
- **route control** -label switching allows the router to make routing decisions using various attributes not only the IP destination address.

## 5 MPLS and traffic engineering

Traffic engineering deals with the performance of a network in supporting users QoS needs. Traffic engineering for MPLS networks involves the measurement and the control of traffic.

The objectives of traffic engineering in the MPLS environment are related to two performance functions [8]:

- **traffic oriented** performance which includes QoS operations.
- **resource oriented** performance objectives which deal with networking resources to contribute to the realization of **traffic oriented** objectives.

The aim of traffic engineering is to find mechanisms to satisfy the growing need of users for bandwidth; thus, the efficient management of the available bandwidth is the essence of traffic engineering.

MPLS plays an important role in engineering the network to provide efficient services to its customers.

RFC 2702 specifies the requirements of traffic engineering over MPLS and describes the basic concepts of MPLS traffic engineering like traffic trunks, traffic flows and LSPs [9]. The advantages of MPLS for traffic engineering include:

- label switches are not limited to conventional IP forwarding by conventional IP-based routing protocols;
- traffic trunks can be mapped onto label switched paths;

- attributes can be associated with traffic trunks;
- MPLS permits address aggregation and disaggregation (IP forwarding permits only aggregation);
- constraint-based routing is easy to implement;
- MPLS hardware is less expensive than ATM hardware.

## 6 MPLS integration with IP and ATM (layer2 and layer3 protocols)

MPLS does not replace the IP protocol. It supplements IP so that traffic can be marked, classified and policed. With the use of MPLS, end-to-end quality of service can finally be achieved [10].

An important aspect of MPLS is its ability to operate as an overlay over ATM and the ability to tunnel IP packets within an MPLS cloud. This makes it possible to integrate IP and ATM operations in a single switch.

ATM and IP are the two most important protocols that Internet forums wish to integrate to improve the performances and capabilities of Internet. On the other hand there are more layer2 or layer3 protocols coexisting. MPLS currently specifies media-specific behavior for ATM, FrameRelay, PPP and Ethernet. Thus MPLS will not be restricted to any particular link-layer.

A goal of MPLS is to be able to support multiple network-layer protocols (hence the name Multiprotocol Label Switching) even if to date, the focus of the effort has been on defining functions and operations for IPv4.

Our project for the moment concerns IPv4 over Ethernet but it may evolve to support other layer2 or layer3 protocols like ATM and IPv6.

## 7 Our project

### 7.1 The idea

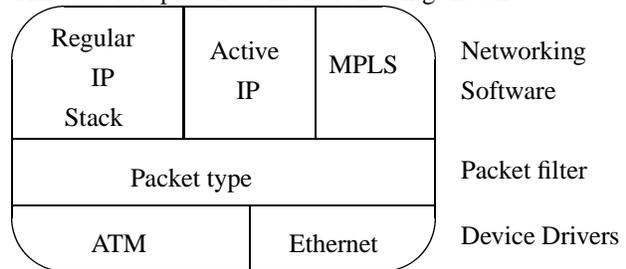
The goal of our project is to develop novel bandwidth management strategies using MPLS. These strategies

will control and distribute the bandwidth available between applications and users which share it. Using bandwidth we intend to verify the performance of these schemes by deploying them in a live MPLS network.

We try to accomplish this by using a software MPLS router/switch. This will act as an universal router/switch by supporting any Layer 2 protocol (ATM, Ethernet, etc.) and by being able to encapsulate any Layer 3 packets (IP, MPLS, Active IP [11], etc).

As the packets arrive to it, they will be filtered and classified. The classification can be made based on media type, destination/source address or based on any other criteria. Those classes of packets will each have a share from the available bandwidth.

This idea is represented in the following sketch:



We are currently working on the implementation of an MPLS switch using a Linux platform. This implementation will help us to test the performance of MPLS. Once we have developed a stable implementation of MPLS we can modify this implementation in order to add in Bandwidth Management capabilities.

### 7.2 Implementing MPLS in Linux

MPLS is not yet a mature standard. There are many aspects of it which still require research. In order to gain a better understanding of MPLS technology and its usefulness to traffic engineering and bandwidth management, we are developing our own MPLS implementation for the Linux operating system. The resulting platform may be used to evaluate alternatives to the proposed MPLS standard implementation and to determine how best to exploit the flexibility of MPLS in bandwidth management.

The standard [4] specifies that the MPLS label should be encoded between layer 2 (PPP, Ethernet)

and layer 3 (IPv4, IPv6) headers of a packet. Another proposal as described in [12] is to encode the label within the layer 2 Media Access Control (MAC) destination address.

Our implementation uses the second approach. This involves work with data-link layer and network layer headers. The research started with understanding ATM and the integration of ATM with Linux. We took into consideration the possibility of developing Linux drivers for ATM network cards. Hence, we studied the Linux protocols and Linux device drivers [13].

Once we understood the IPv4 protocol stack [14] the research continued with implementing new Linux protocols. The modularity of the Linux kernel [15] allowed us to easily implement and integrate a new protocol.

Currently, our *vanilla* MPLS is implemented as a kernel module. This protocol module is independent of the Linux kernel source. Thus it can be loaded and unloaded dynamically from the running kernel. The standard approach requires changes in the internal kernel structures (socket buffers [16]) by adding the label field between the layer 2 and layer 3 headers. This is a major modification since socket buffers are the structures used by the all Linux protocols. Therefore the kernel has to be recompiled and the system has to be restarted.

In our approach we are transparently changing the purpose of an existing MAC field, the destination address field. The MPLS module will use it to push, swap or pop the label. The other components of the Linux networking system will not be affected by the change with one exception. The Ethernet interface must run in promiscuous mode in order for the associated device drivers to receive the modified frames and to push them to the higher level, the MPLS module. Otherwise the Ethernet NIC will reject these frames, since in normal operation it discards frames whose MAC destination addresses do not match its Ethernet address.

This implementation makes the MPLS router hot configurable. By example, a Linux box running as an IP router becomes an MPLS router once we load the MPLS module. If we unload the MPLS module (for debugging or modifications) the system will remain an

IP router. This reconfiguration can be done without recompiling the kernel or restarting the system.

We issued a network test suite for our MPLS protocol and successfully tunneled IPv4 within an MPLS cloud. Two IPv4 stack computers were able to communicate (transparently) using the MPLS cloud as a part of the network core. This demonstrates the usefulness of our approach in, for example, Virtual Private Networks (VPN).

## 8 Conclusions

In this article we presented the work involved in implementing a protocol for next-generation routed networks. By using this protocol, Internet Service Providers will be able to control the allocation of available bandwidth, thereby providing better services and QoS guarantees for Internet users. The next step is to implement a signaling mechanism for the existing forwarding protocol. This will result in a fairly complete implementation of the base MPLS standard. This can then be extended to experiment with new ideas for bandwidth management.

## References

- [1] Solaris. Bandwidth management for ip networks. <http://www.sun.com/software/bandwidth/wp-sbm16/>.
- [2] Eric W. Gray. *MPLS Implementing the technology*. Addison-Wesley, 2001.
- [3] E. Rosen, Cisco Systems, A. Viswanathan, Force10 Networks, R. Callon, and Juniper Networks. Multiprotocol label switching architecture. Technical Report RFC3031, IETF, 2001.
- [4] E. Rosen, D. Tappan, G. Fedorkow, Cisco Systems, Y. Rekhter, Juniper Networks, D. Farinacci, T. Li, Procket Networks, A. Conta, and Transwitch Corporation. Mpls label stack encoding. Technical Report RFC3032, IETF, 2001.
- [5] B. Davie, J. Lawrence, K. McCloghrie, E. Rosen, G. Swallow, Cisco Systems, Y. Rekhter, Juniper Networks, P. Doolan, and Ennovate Networks. Mpls using ldp and atm vc switching. Technical Report RFC3035, IETF, 2001.

- [6] L. Andersson, Nortel Networks, P. Doolan, Ennovate Networks, N. Feldman, IBM Corp, A. Fredette, PhotonEx Corp, B. Thomas, and Cisco Systems. Ldp specification. Technical Report RFC3036, IETF, 2001.
- [7] B. Thomas, Cisco Systems, E. Gray, and Zaffire. Ldp applicability. Technical Report RFC3037, IETF, 2001.
- [8] Uyles Black. *MPLS and Label Switching Networks*. Prentice Hall, 2001.
- [9] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and UUNET (MCI Worldcom) J. McManus. Requirements for traffic engineering over mpls. Technical Report RFC2702, IETF, 1999.
- [10] Rick Gallaher. An introduction to mpls. <http://www.convergedigest.com/Bandwidth/archive/010910TUTORIAL-rgallaha%r1.htm>.
- [11] David L. Tennenhouse David J. Wetheral. The active ip option.
- [12] D. Bussiere (Cabletron), H. Esaki (Toshiba), A. Ghanwani (IBM), S. Matsuzawa (Toshiba), J. W. Pace (IBM), and V. Srinivasan (IBM). Labels for mpls over lan media, 1997.
- [13] Alessandro Rubini. *Linux Device Drivers*. O'Reilly, 1998.
- [14] Stephen T. Satchell and H.B.J. Clifford. *Linux IP Stacks Commentary*. Coriolis, 2000.
- [15] Scott Maxwell. *Linux Core Kernel Commentary*. Coriolis, 1999.
- [16] Herald Welte. skb - linux network buffers. <http://www.gnumonks.org/ftp/pub/doc/skb-doc.html>.
- [17] Mpls for linux project. <http://mpls-linux.sourceforge.net/>.