LAPPEENRANTA UNIVERSITY OF TECHNOLOGY

Department of Information Technology

Master's thesis

MPLS control architecture in a multipurpose switch

The subject of the thesis was approved by the council of the Department of Information Technology on August 15, 2000.

Supervisor: Professor Jari Porras
Instructor: Professor Hannu Kari

Lappeenranta, 24th November 2000

Olli Suihkonen
Haapajärventie 9
77690 Suonenjoki
+358 50 338 3876

ABSTRACT

Author:	Suihkonen, Olli Mikko
Title:	MPLS control architecture in a multipurpose switch
Department:	Department of Information Technology
Year:	2000
Place:	Lappeenranta

Master's thesis. Lappeenranta University of Technology.
74 pages, 13 figures, 2 tables and 2 appendices.
Supervisor: Professor Jari Porras.

Keywords: MPLS, IP, routing, signaling, interworking

The advancement of communication networks has led to the deployment of several differing and incompatible networking technologies. During the last decade, IP has become the most significant data transfer protocol. This has forced the industry to develop new methods for interworking between IP and existing connection-oriented networks.

Using broadband and connection-oriented backbone networks efficiently requires considering the characteristics of IP in data transfer. On the other hand, problems in scalability and service quality, which are typical to IP networks, can be reduced by selectively utilizing the basic features of connection-oriented networks. This work introduces a control software implementation for a broadband network switch using IP routing. In addition, a concept of interworking between an IP-centric backbone network and networks controlled by intelligent network call control is presented. The solution makes it possible to interconnect existing networks widely and to use services over network borders.

This thesis is part of the results for the SCOMS project by the Telecommunications Software and Multimedia Laboratory of Helsinki University of Technology. The switching equipment is developed by the Technical Research Centre of Finland, and it supports cell-based ATM interfaces, n x 64 kbps ISDN and PSTN interfaces and IP-based Ethernet interfaces. The work is focused on IP-centric switch control software that supports interworking with connection-oriented networking technologies.

TIIVISTELMÄ

Tekijä:	Suihkonen, Olli Mikko
Tutkielman nimi:	MPLS-ohjausarkkitehtuuri monikäyttökytkimessä
Osasto:	Tietotekniikan osasto
Vuosi:	2000
Paikka:	Lappeenranta

Diplomityö, Lappeenrannan teknillinen korkeakoulu.
74 sivua, 13 kuvaa, 2 taulukkoa ja 2 liitettä.
Tarkastaja: Professori Jari Porras.

Avainsanat: MPLS, IP, reititys, merkinanto, yhteistoiminta

Tietoliikenneverkkojen kehitys on johtanut useiden erityyppisten ja epäyhteensopivien verkkotekniikoiden laajamittaiseen käyttöönottoon. Viimeisen vuosikymmenen aikana pakettikytkentäinen IP on noussut merkittävimmäksi tiedonsiirtoprotokollaksi. Tämä on pakottanut kehittämään uusia menetelmiä olemassaolevien yhteydellisten verkkojen ja IP:n väliseen yhteistoimintaan.

Laajakaistaisten ja yhteydellisten runkoverkkojen käyttäminen tehokkaasti vaatii IP:n ominaisuuksien huomioimista tiedonsiirrossa. Toisaalta IP-verkoille tyypillisiä palvelunlaatu- ja skaalautuvuusongelmia voidaan vähentää hyödyntämällä yhteydellisten runkoverkkojen perusominaisuuksia valikoidusti. Tässä työssä esitellään toteutus IP-reitityksen käytöstä laajakaistaverkon kytkimen ohjauksessa, sekä konsepti IP-orientoituneen runkoverkon ja älyverkon ohjaukseen perustuvien verkkojen yhteistoiminnasta. Ratkaisu mahdollistaa olemassaolevien verkojen laajamittaisen yhdistämisen ja palveluiden käytön verkkojen rajojen yli.

Työ on osa Teknillisen korkeakoulun tietoliikenneohjelmistojen ja multimedian laboratorion SCOMS-projektia. Kytkinlaite on VTT:n kehittämä ja sisältää tuen solupohjaisille ATM-liitännöille, n x 64 kbps PSTN- ja ISDN-liitännöille sekä IP-pohjaisille Ethernet-liitännöille. Työn painopisteenä ja tuloksena on IP-keskeinen monikäyttökytkimen ohjausohjelmisto, joka tukee yhteistoimintoja edellämainittujen yhteydellisten tekniikoiden kanssa.

PREFACE

This thesis was made in Lappeenranta for the Telecommunications Software and Multimedia Laboratory of Helsinki University of Technology.

I would like to thank my instructor Professor Hannu Kari for his determined guidance through this thesis and for the opportunity to do creative and challenging research work. I am also grateful to Professor Pertti Raatikainen for his collaboration and advice during the project.

Thanks also to the workers on the SCOMS project for creating a pleasant working environment. Fellow students get credit for dragging me away from computers and books to have fun, though I guess it was not usually that difficult.

Suuret kiitokset vanhemmilleni opiskeluaikana saamastani taloudellisesta ja henkisestä tuesta.

Lappeenranta, 24.11.2000
Olli Suihkonen

TABLE OF CONTENTS

ABBREVIATIONS

ABBREVIATIONS

3GPP	3rd Generation Partnership Project
AAL	ATM Adaptation Layer
API	Application Programming Interface
ARIS	Aggregate Route-based IP Switching
ATM	Asynchronous Transfer Mode
BA	Behavior Aggregate
B-ISUP	Broadband ISDN User Part
BGP	Border Gateway Protocol
CC	Call Control
CCF	Call Control Functions
CCM	Cross Connector Mux
CE	Customer Equipment
Diffserv	Differentiated Services
CLIP	Classical IP over ATM
CORBA	Common Object Request Broker Architecture
CR-LDP	Constraint-based Routing LDP
CR-LSP	Constraint-based Routing LSP
CSR	Cell Switch Router
DSCP	Diff-Serv Code Point
DSS1	Digital Subscriber Signaling System No. 1
DSS2	Digital Subscriber Signaling System No. 2
FCF	Fabric Control Functions
FEC	Forwarding Equivalence Class
FIB	Forwarding Information Base
FSM	Finite State Machine
GSMP	General Switch Management Protocol
HDTV	High Definition Television
HUT	Helsinki University of Technology
ICC	Interworking Call Control
IETF	Internet Engineering Task Force

IFMP	Ipsilon Flow Management Protocol
IGMP	Internet Group Management Protocol
Intserv	Integrated Services
IP	Internet Protocol
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
ISUP	ISDN User Part
ITU-T	International Telecommunications Union – Telecom.
LAPD	Link Access Procedures on the D-channel
LDP	Label Distribution Protocol
LLC	Logical Link Control
LSP	Label Switched Path
LSR	Label Switching Router
MIB	Management Information Base
MOSPF	Multicast OSPF
MPLS	Multiprotocol Label Switching
MPOA	Multiprotocol Over ATM
MRT	Multicast Routing Table
MTP	Message Transfer Part
NNI	Network-to-Network Interface
OAM	Operations, Administration and Maintenance
OSI	Open Systems Interconnection
OSPF	Open Shortest Path First
OVOPS	Object Virtual Operations System
PDU	Protocol Data Unit
PE	Provider Edge router
PPP	Point-to-Point Protocol
PSTN	Public Switched Telephone Network
PVC	Permanent Virtual Circuit
QoS	Quality of Service
RFC	Request For Comments
RIB	Routing Information Base

RIP	Routing Information Protocol
RSVP	Resource Reservation Protocol
RTP	Real-time Transfer Protocol
SCC	Switching Call Control
SCOMS	Software Configurable Multidiscipline Switch
SIP	Session Initiation Protocol
SNAP	Subnetwork Attachment Point
SNMP	Simple Network Management Protocol
SSCF	Service Specific Coordinate Function
SSCOP	Service Specific Coordination Oriented Protocol
SVC	Switched Virtual Circuit
TCP	Transmission Control Protocol
TDP	Tag Distribution Protocol
TIB	Tag Information Base
TSR	Tag Switching Router
TOVE	Transparent Object-oriented Virtual Exchange
TTL	Time To Live
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
UNI	User-to-Network Interface
URL	Uniform Resource Locator
VC	Virtual Channel, Virtual Circuit
VCIB	Virtual Channel Information Base
VoMPLS	Voice over MPLS
VPN	Virtual Private Network
VR	Virtual Router
WDM	Wavelength Division Multiplexing

1 INTRODUCTION

The last few decades have seen astonishing progress in the way people communicate. A number of different types of communication networks address the diverging needs of communication services such as telephone calls and data transfer. Landline telephone networks reach everywhere in developed countries. Internet and mobile networks have grown explosively and realized the easy communication between individuals.

Unfortunately, many communication services are specific to the networks in which they are used. Despite easy access to a wide-range of networks, the heterogeneity of them makes it impossible to use all the services between different networks and user interfaces. Since large investments are involved in nation-wide networks, operators are interested in evolving the old-established network infrastructures instead of totally replacing them with modern networking solutions. Convergence of different networks is needed to create a true multiservice network that will bind different networks and services together. Since Internet Protocol (IP) has become increasingly popular, it will be the driving force of convergence.

The telecommunications market research and consulting firm RHK estimates that IP traffic will represent over 90 percent of the total public network traffic by the year 2002. As a result, service providers will need scalable, cost-effective, data-centric backbone networks that can provide guaranteed performance capabilities. [RHK1999]

Internet technologies are also being deployed on mobile networks. Consequently, mobility aspects are becoming more important in backbone networks, and drawing a line between Internet and mobile networks may not be appropriate any longer. The 3rd Generation Partnership Project (3GPP) has announced all IP core networks as its goal in Universal Mobile Telecommunications System (UMTS) technology, in order to offer seamless third generation services. One of the benefits of this approach will be making the services available, independently of the means of access. This requires bringing the existing networks, like conventional land telephony, and future IP networks together [3GP1999].

The next chapter goes through some relevant networking terminology and defines the research problem and the aim of this study. Criteria for evaluating the solution are specified. The third chapter discusses the problems of overlay networks and explains the generic solution for network integration. The fourth chapter presents different proprietary and non-proprietary solutions that have been developed for the integration of IP and connection-oriented broadband networks. The fifth chapter introduces the research project in question and the implementation of the chosen networking technology. The sixth chapter analyses the solution and the fulfillment of the criteria, and the last chapter draws conclusions and sums up the work.

2 PROBLEM

This study involves combining IP and connection-oriented networks, especially broadband backbone networks. Another issue is bringing narrowband connection-oriented networks and IP-based access networks together with the backbone network. Thus, the goal is to research interoperability possibilities between different networking technologies.

2.1 Terminology

2.1.1 General terminology

OSI reference model

Open Systems Interconnection (OSI) is a standard description or reference model for how messages should be transmitted between any two points in a telecommunication network. The main idea of OSI is that the process of communication between two end users in a telecommunication network can be divided into layers, with each layer adding its own set of special, related functions. The OSI reference model has seven layers of which each defines a function performed when data is transferred between applications across a network (Table 1). OSI was officially adopted as an international standard by the International Organization for Standardization (ISO). Currently it is a recommendation of the International Telecommunications Union (ITU-T). This thesis concentrates on layers 2 and 3 of OSI model.

Table 1. OSI reference model.

7. Application	application programs that use the network
6. Presentation	standardizes data presented to the applications
5. Session	session management between applications
4. Transport	error detection and correction, sequence control
3. Network	addressing, routing, message handling
2. Data Link	data delivery across the physical connection
1. Physical	physical network media

Software architecture

Software architecture alludes to ”the overall structure of the software and the ways in which that structure provides conceptual integrity for a system”. In its simplest form, architecture is the hierarchical structure of program components, the manner in which these components interact and the structure of the data that is used by the components. [Pre1997]

Interface

In hardware terminology, an interface means the physical and logical arrangement supporting the attachment of any device to a connector or to another device. An interface in software means a set of messages or signals used between software components. In a networking device, an interface is usually a line card that connects the device to a network.

2.1.2 Switching terminology

Circuit switching

Each connection through a circuit switched network results in a physical communication channel being set up through the network from the calling to the called subscriber equipment. This connection is used exclusively by the calling subscriber and the called subscriber for the duration of the call.

Before any data can be transmitted over such a connection, the connection must be set up or established through the network. An example of a circuit switched network is a Public Switched Telephone Network (PSTN) [Hal1996]. Circuit switching is performed on the data link layer by a device called switch.

Signaling

Signaling in communication networks means exchanging information that sets up, controls, and terminates the connections. Signaling may include several functions, such as negotiating the quality of the connection depending on the network and the user interface.

Switch

A typical switch has two parts: the switching hardware carries user data, and the switch controller handles requests to set up and tear down circuits. The switch uses signaling protocols to manage the connections. It may also have interfaces to exchange control and management information with special purpose networks.

Although a traditional switch performs layer 2 functions, the term has a loose meaning. If a switch performs IP routing functions of layer 3, it may be called an IP switch. Layer 4 switches have been introduced to make even smarter forwarding decisions. Sophisticated multilayer switches are able to switch simultaneously different traffic in layers 2, 3 or 4.

Virtual Circuits

A virtual circuit is a circuit or path between points in a network that appears to be a discrete, physical path but is actually a managed pool of circuit resources from which specific Virtual Circuits (VC) are allocated as needed to meet traffic requirements. A Permanent Virtual Circuit (PVC) is a VC that is permanently available to the user just as though it were a dedicated or leased line continuously reserved for that user. A Switched Virtual Circuit

(SVC) is a VC in which a connection session is set up for a user only for the duration of the connection [Wha2000].

The biggest advantage of virtual circuits is that they do not reserve bandwidth when a connection is idle. However, when circuit switching is mentioned in this document, it may apply to both hardware circuits and virtual circuits, because the differences between these two are not relevant in this context.

Interworking

Interworking between different networks and between different signaling protocols implies mappings and information conversions between the communicating parties. Both the hardware platform and the software framework have to implement different signaling protocol stacks with different signaling procedures and be able to interconnect them. In addition, different kinds of resource allocations and verifications, like virtual circuits and time-slots, have to be provided [Raa1999a].

Multipurpose switch

A multipurpose or multiprotocol switch is a device that is capable of interconnecting different types of networks and protocols. Interworking functions are implemented to handle each type of interconnection separately, and the switch may perform data conversions between the different types of media.

2.1.3 Routing terminology

Packet switching

With a packet switched network no physical connection is established through the network. Instead, all data to be transmitted is first assembled into one or more message units, called packets. These packets include both the source and destination network addresses. At a packet switching exchange a packet is forwarded on an appropriate link, based on the destination address.

Each overall transmission occupies only a portion of the available bandwidth for each link, since packets from different sources are interspersed. Single packets may experience unpredictably long delays when the network is congested [Hal1996]. IP is the single most important packet switching protocol. Packet switching is performed on the network layer by a device called a router.

Router

A router is connected to at least two networks and decides which way to send each packet based on its current understanding of the state of the networks it is connected to. A router creates or maintains a table of the available routes and their conditions and uses this information along with distance and cost algorithms to determine the best route for a given packet (Figure 1).

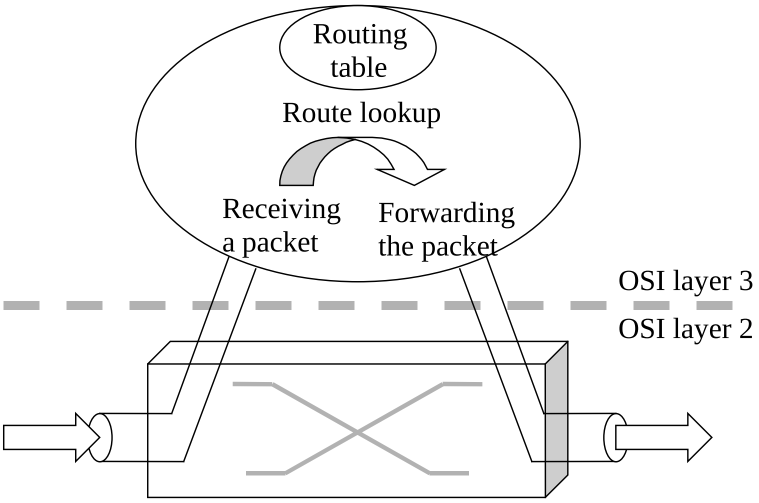


Figure 1. Conventional routing procedure [Pet1996].

Typically, a packet may travel through a number of routers before arriving at its destination. A router is often included as part of a network switch

[Wha2000]. In general, a router is a more complicated and slower device than a layer 2 switch.

Dynamic routing

Dynamic routing occurs when routers talk to adjacent routers, informing each other of what networks each router is currently connected to. The routers must communicate using a routing protocol, of which there are many to choose from [Ste1994].

Current IP routing protocols such as Open Shortest Path First (OSPF), Routing Information Protocol (RIP) and Border Gateway Protocol (BGP) are best effort routing protocols. They calculate the shortest path to the destination, not on basis of shortest physical distance but the amount of hops, cost or delay. An IP packet is routed through a path that is considered the best possible route between the source and the destination.

2.1.4 Traffic terminology

Quality of Service

The quality of the transmission path can be measured with varying characteristics, like bandwidth, latency, delay variation, reliability, and other requirements. The idea of Quality of Service (QoS) is that these characteristics can be guaranteed in advance, measured and improved.

Real-time traffic

Different services have different QoS needs. Time-critical ones, such as telephone calls or streaming video, suffer greatly from delay and delay variation. They also need a constant bandwidth to work properly. These services work well on circuit-switched networks, since the concept guarantees a high-quality transmission path for the call [Bel2000].

Non-real-time traffic

Typical services used on the Internet, like browsing the web, produce data bursts. Using (hardware) circuit-switched connections for non-real-time traffic causes wasted capacity, because the reserved bandwidth is not used continuously. Packet-switched networks are ideal for these services, because they acquire and release the bandwidth, as it is needed. The weakness of IP is that it offers universally only a single class of best effort service; that is, the network offers no assurance about when, or even if, packets will be delivered [She1994]. QoS methods for IP have been developed, but their deployment has so far been slow and occasional.

2.2 Integration of IP and connection-oriented networks

Interworking between connection-oriented networks such as PSTN and Integrated Services Digital Network (ISDN) is straightforward from the controlling viewpoint. A multipurpose switch creates connections from one network to another with the use of network specific signaling protocols. Different signaling protocols have usually similar sequences of messages; therefore mapping parameters from one type of protocol to another is sufficient. The switch knows how to route an incoming call with a certain telephone number or network address.

Interworking between IP and connection-oriented networks is difficult because of the different data transfer models. Interworking between IP and broadband networks means carrying IP traffic over a layer 2 broadband transport media in a way that considers the characteristics of IP. This requires introducing a new network control mechanism when operating with broadband technologies that are originally designed for telecommunication networks. Implementing router functions to a layer 2 switch is necessary but insufficient since connections are opened on layer 2 and routing is done on layer 3.

QoS characteristics are usually specified in the signaling protocol parameters and therefore easily available. The problem is how the requirements can be applied to the network that carries IP. Transferring data with IP, using connection-oriented services is difficult. If a voice call is opened over IP, a specific signaling protocol like Session Initiation Protocol (SIP) or a connection-oriented protocol like Real-time Transfer Protocol (RTP) is necessary even to set up the call. Moreover, even with the use of a dedicated protocol, the underlying packet switched IP that provides only best effort service makes it impossible to guarantee adequate QoS. Retransmission of lost or corrupted packets is not the answer for the needs of real-time services.

Layer 2 switching can offer guaranteed QoS, but running connectionless IP on top of it wastes the advantages of connection-oriented switching. IP is the answer to non-real-time traffic and circuit switching is to real-time traffic. However, we need to be able to transfer all types of traffic with different QoS requirements on the same network. This can be achieved by integrating the operations of layers 2 and 3 together and combining the best of them.

2.3 Problem statement

The aim of this thesis is to develop a control software architecture for a multipurpose switch to enable forwarding IP-based real-time and non-real-time traffic. Two goals are especially taken into account:

- Layer 3 routing information must be utilized efficiently when opening layer 2 connections in a backbone network.
- Real-time services must be supported with the use of signaling, resource reservation, and interworking functions with other network types.

An introduction to different multilayer technologies is presented, and interworking issues between different types of access and backbone networks are discussed. The emphasis is on a single multiprotocol switch and its functionality. The IP routing information of the IP-based backbone and access networks is regarded to be available in a routing table.

The following aspects concerning the problem were considered to be outside the scope of this study:

- user data conversions between different networks
- the exact role of the switch inside a network
- possible advancements of routing protocols and layer 3 packet handling algorithms
- upcoming new transport technologies like Wavelength Division Multiplexing (WDM) and Gigabit Ethernet

This work is about advancing current layer 2 switching equipment of backbone networks by applying new IP-centric technologies to them. This may be a considerable option for operators who seek a fast and cost-effective way to improve QoS and extend the functionality of their current networks. These networks are also test benches for new solutions, and the gained experience should be utilized when designing future networks.

2.4 Criteria

The following criteria can be used in evaluating different solutions for the problem statement.

Functionality

- Fine/coarse-grained QoS support for static and dynamic needs
  = *Creating paths with different service characteristics through the network both statically and dynamically.*

- Interworking ability with other communication networks
  = *Making interconnections between different protocols and types of user data.*
- Traffic engineering facilities
  = *Optimization, measurement, modeling, characterization and control of traffic.*
- Virtual Private Network (VPN) support
  = *Connecting a set of physically separate sites of network securely.*
- Multicasting capability
  = *Delivery of the same data to multiple destinations and the support for a multicast routing protocol.*

Applicability

- Scalability
  = *Being able to scale, especially when operating on large backbone networks.*
- Independence of the routing protocol
  = *Any layer 3 routing protocol may be run on the device.*
- Generality with respect to link layer technologies
  = *Solution is applicable to any layer 2 technologies of today and of the future.*
- Portability and readiness for future updates
  = *The solution may be transferred to other operating environments and updated when necessary.*

Controllability

- Operations, Administration and Maintenance (OAM) facilities
  = *Controllability over the functionality of the device.*
- Minimal complexity, modular architecture
  = *Logical and clear arrangement of software modules.*

Performance

- Control information handling does not delay user data
  - = *The layer 2 technology is used at maximum speed.*

3 GENERIC SOLUTION

This chapter contains a generic solution for networking with IP and layer 2 switching technologies. A traditional approach is first introduced to clarify the purpose of the integration of layers 2 and 3, before discussing a generic solution for the integration. Discussed techniques are also applicable to other layer 3 packet switched protocols other than IP.

3.1 Classical overlay model

The overlay model is a technique that was deployed within the 1990s to circumvent some of the limitations of IP systems. The basic idea is to introduce a secondary technology, with switching and traffic management capabilities into the IP infrastructure in an overlay configuration. The switched connections of the secondary technology serve as point-to-point links between IP routers as illustrated in figure 2 [Awd1999a].

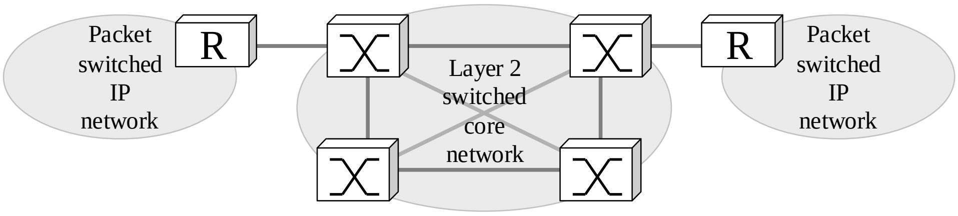


Figure 2. Routers and switches in an overlay configuration.

The routers that are attached to the core network are directly connected to other routers by layer 2 connections that are usually pre-configured and thus static. IP routing is limited to the edges of the network. While this method has proven to be very effective, operational and technical difficulties begin to arise since the number of connections grows exponentially as the network expands [Fry2000].

Since the number of network devices may be very large, the next-hop routing tables can become unmanageable. For n routers in the network, the routing table size, the routing update traffic, and the processing of routing updates all grow to $O(n^2)$. A conventional solution is to leave some of the possible paths in the network unconnected. Thus, packets between two physically adjacent routers may be required to traverse multiple hops. This is inefficient and it is difficult to maintain connectivity when links or routers fail [New1998].

The most challenging problem has been the complexity of operating a network based on two disparate technologies that were independently designed and developed for entirely different tasks. Figure 3 illustrates the edge of an overlay network, and the equipment required for network operation. Outside the core network layer 3 routing and forwarding mechanisms are utilized for packet delivery, while inside the core network data is forwarded with the use of a layer 2 transport mechanism. IP routing is delivered through the core network, but it is not associated with layer 2 connections.

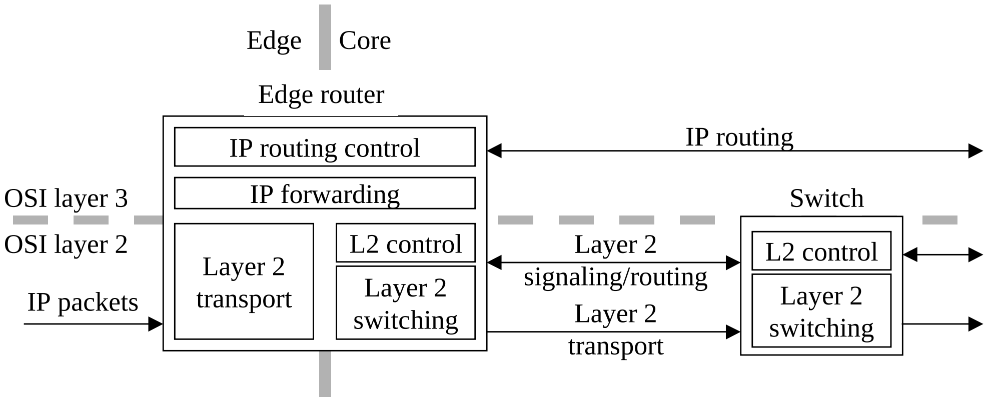


Figure 3. Edge and core equipment of an overlay network.

Different protocol architectures, addressing models, routing protocols, signaling protocols, and resource allocation schemes for both layers make it very difficult to construct an efficient network [Sem1999]. This has forced the hardware vendors and operators to search for different solutions that do not require two separate sets of equipment. The trend is to evolve core IP networks away from the overlay model and toward more integrated solutions [Awd1999a].

3.2 Multilayer switching

Multilayer switching means combining traditional layer 2 switching with layer 3 protocol routing in a single product. Since the mid-1990s, when the shortcomings of the overlay model became obvious, the area has been actively researched and several varying solutions have been introduced.

3.2.1 Overview

Different multilayer technologies have two main characteristics in common (Figure 4):

- separation of the control and forwarding components
- label-swapping forwarding algorithm [Sem2000]

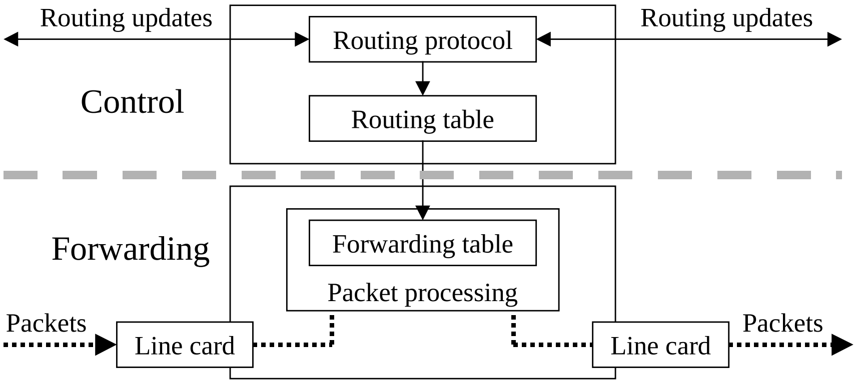


Figure 4. Multilayer components.

By completely dividing the control component from the forwarding component, each component can be independently developed and modified. The division is also between the urgency of information. Forwarding user data needs very fast decision making on hardware, while control information may be processed on a software component that is slower but makes more complicated decisions.

The basic approach of multilayer switching is to take the control software from an IP router, and integrate it with fast layer 2 switching hardware. As a result, it is possible to build a device that has the price and performance of a switch, but the functionality of a router. [Anr2000] This approach also allows the vendors to reuse existing software and hardware solutions.

The forwarding component is based on a label-swapping algorithm. Usually, the same algorithm is used on normal layer 2 switching. A label is a short, fixed-length value carried in the header of a packet to identify a group of IP packets that are handled alike. The labels are analogous to layer 2 connection identifiers. [Sem2000] Multilayer switching is also called label switching.

The routing information is mapped into a forwarding table that contains information about incoming and outgoing labels and interfaces. There may be one forwarding table per router or one per interface, depending on the implementation. The forwarding algorithm can be very simple; it simply takes a packet, finds an entry in the forwarding table on the basis of the incoming label and the interface, then changes the label and forwards the packet on the correct interface.

3.2.2 Basic QoS techniques

There are two basic approaches when trying to guarantee QoS. They are not mutually exclusive, but complementary and designed for varying operational requirements in different networks.

Resource reservation (Integrated Services):

Network resources are apportioned according to an application's QoS request, and subject to bandwidth management policy. This approach requires signaling through the network in order to deliver the QoS criteria to each network device that performs the reservation of resources. Integrated Services (Intserv) is a fine-grained QoS method.

Prioritization (Differentiated Services):

Network traffic is classified and apportioned network resources according to bandwidth management policy criteria. The QoS criteria are carried with the user data, and the classifications give preferential treatment to applications that have requirements that are more demanding. [Sta1999] Differentiated Services (Diffserv) is a coarse-grained QoS method since only a small number of traffic classes are used.

3.2.3 Multilayer switching in solving the QoS problem

Multilayer solutions belong to the resource reservation category, but they may also utilize prioritization. The difference is at which point in the network the action is taken. Layer 2 connections are established on the basis of some information that is available on the edge of the multilayer network. This information may be obtained from routing or signaling protocols, or carried with the user data. When a flow of IP traffic is forwarded between two switches, it is said that it proceeds from an upstream switch to a downstream switch. Therefore, a label switched connection is unidirectional. A router on the edge of the multilayer network is called an ingress for a certain label switched connection if the data arrives through it, or an egress if the data leaves through it.

If the creation or termination of a connection is triggered by data packets arriving at the multilayer switch, the technique is referred to as data-driven connection management. A control-driven model manages the connections with the use of control information such as signaling or routing. With control-driven approach, the connections are created before the first packets of user data arrive, while with data-driven approach each data flow is identified before opening connections [Sem2000].

When a label switched connection is opened, it fulfils some QoS criteria that is available for the layer 2 technology. By then resources are reserved for the data flow on layer 2 and the information of layer 3 is no longer checked. The label of a packet contains information sufficient for forwarding the packet, and may contain information that indicates what resources the packet can use.

The prioritization of user data must be enabled on the basis of upper layer information, usually meaning that the consideration is done on the edge of the multilayer network. It is possible to open separate switched connections for different priorities, and choose a correct connection for each packet. A queuing mechanism can also be implemented for packets, but queuing packets on the edge router has limited effect to the whole transmission.

3.2.4 Multilayer edge router functionality

The different cases of interconnections, that a multipurpose switch must be able to perform when it operates as an edge router for a multilayer network, may be separated. It is assumed that the switch has IP interfaces that are used to connect IP-based access networks to the backbone network. In order to have reasonable packet forwarding rate, the IP interfaces should have routing capabilities in the hardware. Multilayer interfaces have a forwarding table that enables delivering packets to the right connection, assuming that the connection identifiers are set on the outgoing interfaces. Actions of the edge router depend on the characteristics of a particular multilayer technology, but some common outlines can be drawn.

Two IP interfaces

A packet is transferred between two access networks. A route lookup is performed, and the outgoing interface is obtained. An outgoing IP interface may also perform another route lookup to obtain an access network specific hardware address.

Two multilayer interfaces

Data is transferred from one multilayer interface to another. This is similar to normal layer 2 switching, and layer 3 functions are not performed in forwarding operation. The connection has been opened beforehand with the use of routing information and a signaling operation. Alternatively, a data-driven connection establishment may be performed.

IP and multilayer interfaces

When a packet arrives at an IP interface, a route lookup is done and the next hop is identified as a multilayer type. Connections on a multilayer network are established on basis of IP routing; therefore when using the control-driven mode, a connection exists and the packet may be forwarded to it. Possibly, IP QoS characteristics are taken into account when choosing the correct connection. In the data-driven mode, the data is sent using a hop-by-hop method that is based on layer 3 (or higher layer) information, and the label switched connection is opened along the route.

4 MULTILAYER SWITCHING EFFORTS

This part describes different multilayer switching solutions from the viewpoint of the controlling software. Three proprietary technologies by well-known hardware vendors are introduced to demonstrate different possible approaches, and then a standard-based technology is explained. Common link layer technology for all the discussed solutions is Asynchronous Transfer Mode (ATM) that is next outlined briefly.

4.1 Basics of ATM

In the early 1990s, a new technology called ATM, being standardized by ITU-T, started to catch attention. ATM technology was developed to address the needs of new telecommunication services, such a video-on-demand, video conferencing, high-speed data transfer, videophony, home education and shopping, teleworking and High Definition Television (HDTV), that were predicted to be the wave of the future. [Pry1995] In addition to ITU-T, an industry consortium, ATM Forum, was established to accelerate the deployment of the new technology.

ATM is a packet-oriented technology based on asynchronous time division multiplexing. ATM uses fixed length (53 bytes) cells, of which 48 bytes are reserved for information field and 5 bytes for header. The header is used to identify cells belonging to the same virtual channel and thus used in appropriate routing. Because small cells can be handled quickly, a virtual channel may be used the same way that the circuits of a circuit-switched network. Therefore, a virtual channel may also be called a virtual circuit [Gru1996].

ATM connections may be PVCs that are permanent and created on the basis of configuration, or SVCs that are opened dynamically with the use of signaling. Each virtual channel is given a set of parameters that indicate the

QoS requirements of the connection. One of the main advantages of ATM is that the connections can be tailored to address the needs of customers [Gru1996].

4.2 Proprietary multilayer switching solutions

The growth of the Internet caused wide deployment of the IP-over-ATM overlay model. Several Requests for Comments (RFC) concerning IP encapsulation and delivery over ATM networks [Gro1999, Lau1998, Per1995] have been produced by the Internet Engineering Task Force (IETF), but they do not solve the previously discussed problems of the overlay model. Different multilayer approaches have emerged to do this, and while they are mostly developed especially for ATM, the basic ideas are applicable to other link layer technologies as well.

4.2.1 Ipsilon IP Switching

IP Switching was one of the first multilayer technologies, announced by a start-up company Ipsilon in 1995. The technology caught a lot of attention during 1996-1997 because of interesting approach, combined with powerful and daring marketing. Ipsilon's premise was that ATM and the standards that surround it are too complicated and address wrong problems. Ipsilon decided to go its own way by throwing away the standards of the ATM Forum. ATM Forum was seen as an organization where only ATM technology is important and classical protocols are disregarded. Ipsilon took the point of view that the data transfer must be done in terms of IP, and ATM is merely a fast transport mechanism [Wit1997].

The basic idea of the IP Switching is to remove the software resident in the control processor of an ATM switch, and replace it with standard IP routing software that performs IP routing and forwarding. The IP switch control component in Ipsilon's products runs on an Intel-based workstation that is connected to the switch. The control component also contains extensions that allow it to make use of the switching hardware. These extensions include Ipsilon Flow Management Protocol (IFMP) to associate IP flows with ATM virtual channels, a flow classifier to decide whether to switch each flow and General Switch Management Protocol (GSMP) to control the switch hardware [New1997, Dav2000].

IP Switching is a data-driven label switching approach, since it binds single IP flows with labels by the time the flows are recognized. Virtual channels of ATM represent labels, and the process of label binding is described with the term flow redirection. When an IP Switch is making a decision about label switching a flow, the flow is forwarded through a default VC that leads to the IP Switch control component (Figure 5). The controller routes a predefined amount of IP packets through the outgoing default VC before it decides that the flow is long enough to be label switched. After that, the IP Switch tells its upstream neighbor to redirect a particular flow to it using a specific label. Each downstream IP Switch on the route makes the decision of label assignment independently of other switches.

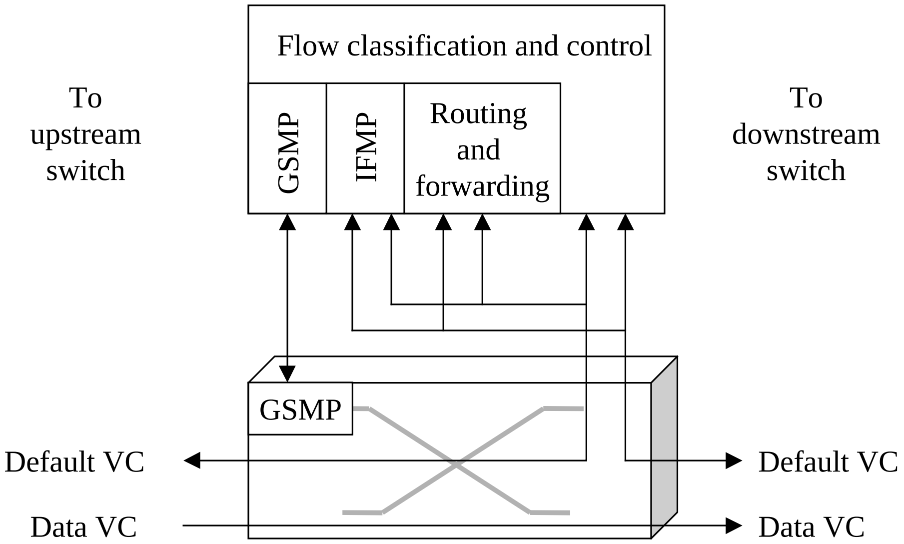


Figure 5. IP Switch architecture [Dav2000].

IP Switching was a pioneering technology, and the first multilayer technology that actually made it into real products. However, it never achieved great success in the marketplace. One of the biggest reasons for this was the performance concern about the data-driven label binding approach, because in IP Switching parts of user data flows must be forwarded through the control component of each switch on the path. The amount of control traffic is directly proportional to the number of traffic flows; therefore, short-lived flows can impose a heavy burden on network operations. While IP Switches were being made available in 1997, the industry was already looking into new promising control-driven solutions. Ipsilon was acquired by Nokia in late 1997 and thereafter IP Switching has been quietly buried. Nowadays, the common understanding is that the data-driven approach in general does not have scaling properties required for operation in the core of the Internet [Dav2000, Sem1999].

4.2.2 Cisco Tag Switching

Cisco announced its control-driven multilayer solution, Tag Switching, in 1996. An effort for standardizing the technology was also initiated with the publication of several Internet drafts. Labels of the scheme are called tags, and the protocol used for delivering them inside the network is Tag Distribution Protocol (TDP). A Tag Switching device is called a Tag Switching Router (TSR). The design goals of Tag Switching are broad. In addition to the delivery of IP routing, the focus is on adding functionality (such as explicit routes) and improving scalability of the network [Dav2000].

Tag switching bases on the concept of binding between a tag and layer 3 routing. Tag Switching supports a wide range of forwarding granularities in order to improve scalability. A tag could be associated from a group of routes to an individual application flow. A tag could also be bound to a multicast tree [Rek1997a].

The control component of a TSR is represented in specifications as a collection of software modules; each designed to support a particular routing function. The software modules are:

- **Destination-based routing module** is used for creating a binding between an outgoing tag and a route, and updating the forwarding table on hardware with the binding information.
- **Hierarchy of routing knowledge module** is used for keeping the routers of a domain from maintaining exterior routing information.
- **Explicit routes** override the destination-based routing paths. Applications may include having finer control over traffic distribution over multiple links and supporting QoS-based routing.

Resource reservation.

Multicast module may be used for handling a multicasting routing protocol and multicast routes [Rek1997b].

TSR also contains a set of information bases. Two of them are mandatory to support Tag Switching:

Forwarding Information Base (FIB) that is maintained by routing protocols and is a normal router database.

- **Tag Information Base** (TIB) that corresponds to the forwarding table of multilayer switching. A TSR allocates tags and binds them to address prefixes in its FIB. Tag information is delivered to neighbors with TDP [Rek1997b].

The problem with Tag Switching is that the Protocol Data Units (PDU) from different sources destined for a specific destination end up sharing a VCI, and cells may get interleaved, which is a problem with ATM [Amo1998].

Tag Switching products were delivered, and Cisco published a set of Internet drafts, initiating also the standardization of multilayer switching, proposing Tag Switching as the base for the new non-proprietary technology to be developed. Tag Switching as such did not make it into a non-proprietary standard, but it had great influence on the upcoming multilayer switching standardization work of IETF.

4.2.3 IBM Aggregate Route-based IP Switching

Late in 1996, shortly after Cisco’s announcement of Tag Switching, IBM's Networking Hardware Division submitted its control-driven Aggregate Route-based IP Switching (ARIS) proposal to the IETF. At this point, after several multilayer solutions had appeared during a short period, interest for a common non-proprietary solution was growing.

ARIS, a simple virtual channel establishment protocol, enables Layer 2 switching of IP packets. It takes advantage of the speed and cost points of ATM switches by prebuilding trees that allow IP packets to be switched to their destination instead of routed. ARIS achieves switching speeds for IP forwarding without affecting the scalability of router networks. Because an ARIS network works like an ordinary IP network, an existing IP network can be upgraded to higher speeds without redesigning the network or changing operational procedures or network management tools [Mar1997].

These switched paths may have an endpoint at a directly attached neighbor (comparable to IP hop-by-hop forwarding), or may have an endpoint at a network egress node, enabling switching via all intermediary nodes. A switched path is created for an egress identifier, which identifies a routed path through a network. The egress identifiers may be extracted from information existing in the routing protocols, or may be configured. Since thousands of IP destinations can map to the same egress identifier, the number of switch paths required in a network is minimized [Fel1997, Vis1997].

The switched path to an egress point, in general, takes the form of a tree because of the merging of switched paths. Merging occurs when multiple upstream switched paths for a given egress point are spliced to a single downstream switched path for that egress point. Generally, merging of ATM virtual channels requires special support from the ATM hardware to avoid the interleaving of cells that belong to different upper layer frames [Vis1997].

From a software viewpoint, ARIS architecture requires two control protocols and three logical information bases. A standard IP routing protocol such as OSPF or BGP is used for obtaining the network topology. ARIS protocol is used for establishing switched paths through the network. The control component includes the following information bases:

- **Routing information bas**e (RIB) is used for the computation of best effort routes by an IP routing protocol. The RIB is essentially unchanged from the RIB of a standard router. In the ARIS context, the RIB is also used to identify egress points and egress identifiers for the other two information bases.
- **Forwarding information base** (FIB) of a standard router has been extended to include an egress identifier in each next hop entry. The FIB tends to contain many IP destination prefix entries, which point to a small number of next hop entries that describe the hop-by-hop forwarding operation(s). Next hop entries consist of an outgoing interface, next hop IP address, egress identifier, and the associated established downstream label for the switched path. The association of the next hops with the egress identifiers is the responsibility of the routing protocol, while the association of the next hop/egress identifiers with the switched paths is the responsibility of the ARIS protocol.
- **Virtual channel information base** (VCIB), which does not exist on a standard router, maintains for each egress identifier the upstream to

downstream label mappings and related states. This mapping is controlled by the ARIS protocol [Vis1997].

IBM planned to extend its previous ATM switching solutions to support ARIS. The proposals IBM has announced have interesting ideas, such as creating loop-free path trees with VC aggregation and they contain considerations of multicasting and routing with multiple paths. The ARIS protocol specification is detailed containing even pseudo code examples. However, in 1997, development of Multiprotocol Label Switching (MPLS) had begun the different proprietary solutions have not achieved great success or interest. Nevertheless, the proprietary multilayer switching technologies initiated the development of MPLS and shaped its form.

4.3 MPLS

A number of other proprietary multilayer solutions has emerged as well. Toshiba has developed a Cell Switch Router (CSR), and Cascade has developed an IP Navigator. Several ATM vendors support Multiprotocol over ATM (MPOA) that is a more ATM-centric approach. Next Hop Resolution Protocol (NHRP) was driven by 3Com, IBM and Cascade to consolidate their IP switching technologies. It is obvious that with all these different competing technologies, the problems for which the idea of multilayer switching had been developed could not be solved universally. Since all the multilayer solutions have common characteristics, it was important to initiate a standardizing effort that would combine the best of each proprietary technology.

4.3.1 Overview

MPLS working group of IETF was found in the spring of 1997. Standardization of MPLS has been slow, but the technology has gained considerable attention since currently (July 2000) 27 Internet drafts have been published by IETF and several dozen as independent drafts. In addition, MPLS is the preferred technology in ITU-T's draft recommendation [IpA1999] that concerns the transport of IP over ATM-based public networks. An industry consortium called the MPLS Forum has been established to accelerate the adoption of MPLS and its associated technologies. Since MPLS has caused so wide interest, it is safe to say that it is one of the most important IP-centric broadband networking technologies of the near future.

The MPLS working group has set high level requirements for their standardization work. The solution must work with existing data link technologies and routing protocols. It should allow a wide range of forwarding granularities associated with a given label, from a single application flow to a group of topologically related destinations. It must also address working in hierarchical networks and address scalability issues. The solution is expected to include protocols for the distribution of labels between routers, encapsulations, multicast considerations, use of labels to support higher layer resource reservation and QoS mechanisms, and definition of host behaviors [Mpl2000].

The fundamental concept of MPLS bases on the ideas of proprietary multilayer solutions, such as the ones discussed before. The technology allows topology-based, control-driven (Tag Switching and ARIS), data-driven (IP Switching) and request-based (Intserv) operation. It is applicable to any layer 3 protocol. The initial standardization effort, however, concentrates on IP version 4. Concerning different layer 2 technologies ATM has gathered most attention, and therefore the Internet drafts mostly take ATM as an example when discussing the operations of MPLS networks. Label distribution is possible with different protocols, such as Resource Reservation Protocol (RSVP), or labels may be piggybacked on a routing protocol. MPLS standard also specifies Label Distribution Protocol (LDP) that is an MPLS specific protocol for the task [Ros1999a].

4.3.2 Label Distribution Protocol

LDP associates a Forwarding Equivalence Class (FEC) with each Label Switched Path (LSP) it creates. The FEC associated with an LSP specifies which IP packets are mapped to that LSP. LSPs are extended through a network as each Label Switching Router (LSR) swaps an incoming label with an outgoing label assigned to the next hop for the given FEC. Two LSRs that use LDP to exchange label/FEC mapping information are known as LDP peers with respect to that information. An LDP session is established between them to exchange the information. A single LDP session allows each peer to learn the other’s label mappings; therefore, the protocol is bi-directional [And2000].

LDP uses connectionless User Datagram Protocol (UDP) for sending "Hello" messages periodically to other LSRs. When an LSR learns about another LSR via Hello messages, and decides to open an LDP session with it, it starts an LDP initialization procedure, of which messages are sent over connection-oriented Transmission Control Protocol (TCP) transport. When a connection is established, LSPs are peers and can start exchanging label binding information [And2000].

One important characteristic about MPLS is that it is possible to open LSPs based on also other constraints than IP routing. Constraint-based routing offers the opportunity to open LSP based on explicit route constraints, QoS constraints and other constraints. These requirements may be met by extending LDP for support of constraint-based routed LSPs (CR-LSP). This makes it possible to have an end-to-end setup mechanism of a CR-LSP initiated by the ingress LSR. An LDP extension for this is called a Constraint-based Routing LDP (CR-LDP), and it specifies explicit routes, traffic parameters, CR-LSP priorities and other requirements [Jam2000].

5 IMPLEMENTATION

This chapter contains a description of the main features of the research project, which this thesis is a part of. Then the standard-based implementation of MPLS and its relation to already existing software components are explained.

5.1 SCOMS project

The Software Configurable Multidiscipline Switch (SCOMS) was initiated in 1998 by Technical Research Centre of Finland (VTT, Valtion Teknillinen Tutkimuskeskus) and Helsinki University of Technology (HUT). The project aims to search possibilities for interworking between different networking technologies and protocols with the use of a multipurpose switch. The project finishes at the end of 2000 [Sco2000].

5.1.1 Overview

The SCOMS project focuses on developing a hybrid switching and routing solution for cell-based, n x 64kbps and packet traffic. VTT is developing the physical switch, and HUT is developing distributed and object-oriented signaling and control software. SCOMS extends the work of the Transparent Object-Oriented Virtual Exchange (TOVE) project that introduced an open and standard-based infrastructure for broadband ATM networks [Pär1999]. Further research has brought standard-based support for narrowband ISDN and PSTN networks.

SCOMS signaling software is located at a separate workstation that runs Linux and is connected to the switch via ATM. The signaling channels of physical network interfaces are switched to the control workstation, as illustrated in Figure 6. One VC per interface is reserved for transferring signaling PDUs that are processed in layered signaling protocol stacks of the control software. Protocols decode and encode messages, use Finite State Machines (FSM) to handle message sequences, and are connected to Interworking Call Control (ICC) to enable the connections between networks. Signaling software architecture, details of used acronyms and signaling protocols are further discussed in [Raa1999a, Raa1999b, Raa1999c and Raa2000].

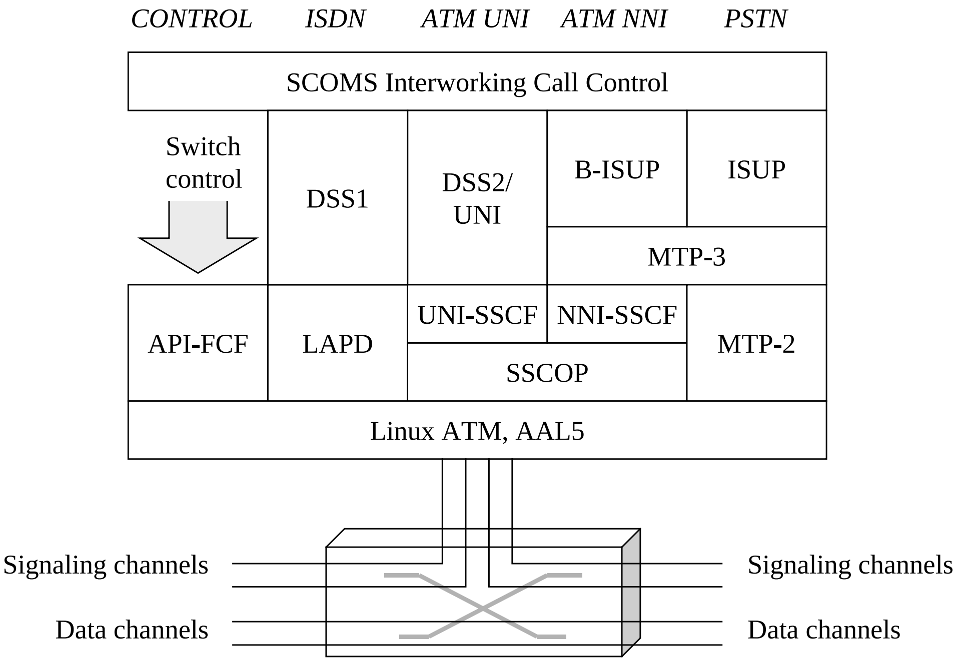


Figure 6. SCOMS signaling software architecture.

The last addition to the control software during the SCOMS project was the support for IP. MPLS was chosen for the IP-centric broadband technology to continue the use of well-known networking standards in order to achieve as good interoperability as possible. We discuss the IP and MPLS software solution and signaling interworking issues of SCOMS in [Sui2000b].

5.1.2 SCOMS software architecture

Different modules of the SCOMS signaling software, for example the signaling protocols, call control and fabric control are independent components. Component-based approach makes it possible to create a flexible software structure that can be extended when needed. All the components are independent, but use the information about the functions of other components to reduce the extra work [Raa1999a].

The interworking signaling implementation consists of software modules, compiled separately and linked together to form an entire software package. The major modules are the ICC module and the signaling stack modules (Figure 7). The ICC module includes the basic and generic Call Control (CC) and Switching Call Control (SCC) modules. The CC module implements a Service Switching Point (SSP) which offers different call control functions (CCF), basically call models, for the originating (CC-O) and terminating (CC-T) sides of a call. The SCC module interconnects CC instances and provides interworking functions and protocol specific bearer connection control functions. The Fabric Control Functions (FCF) module includes all the functions needed to control physical hardware connections. FCF utilizes the Application Programming Interface (API) of the physical switch [Raa2000a].

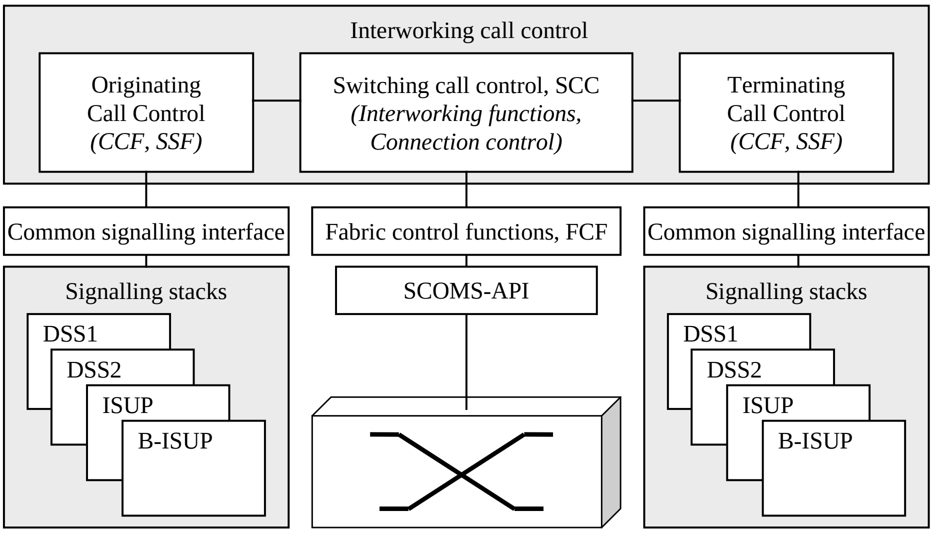


Figure 7. Signaling software modules [Sui2000b].

5.1.3 OVOPS++

SCOMS software has been written in C++, using a tool called Object Virtual Operations System ++ (OVOPS++). It is an object-oriented protocol framework for implementation of data communication software, and has been developed in TOVE and SCOMS projects. OVOPS++ is based on integration of Conduits+ [Hün1995] that is a component-based model of a framework, and OVOPS framework that solves general problems appearing in protocol implementation. OVOPS++ takes advantage of object-oriented design patterns and is therefore well suited for component-based software development.

OVOPS++ provides base classes that are used to derive protocol specific classes. There are two main sets of base classes: conduits and information "chunks". Conduits are channels that are used for transferring and processing information chunks, and each conduit has an FSM to vary its functionality depending on its current state. Conduits include base classes for protocol, multiplexer, factory and adapter. They are utilized for creating layered software architectures, such as protocol stacks as illustrated in

Figure 8. The framework also provides a scheduler and timers for protocol functions. Protocol specific messages are carried in transporters to enable their delivery between any two conduits.

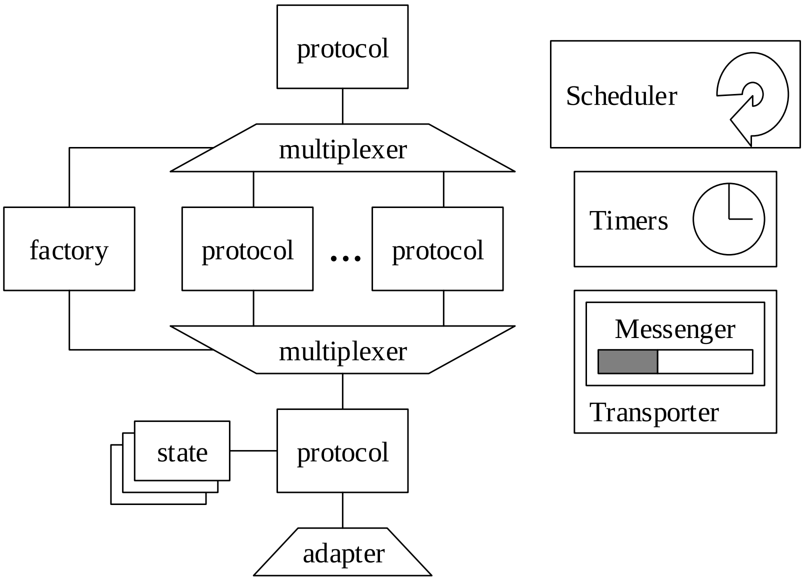


Figure 8. OVOPS++ components.

Each conduit, except adapter, has two sides (A and B) to which other conduits can be connected. Since each conduit with its own characteristics can be attached to any of the other conduits, OVOPS++ makes it possible to construct complicated structures with both static and dynamic parts [Sui2000].

5.2 MPLS implementation in SCOMS

Adding IP functionality to the SCOMS environment requires constructing a new software architecture. The basic function of multilayer switching, mapping IP routing to the ATM level, requires the use of a “signaling protocol” which in the case of MPLS is LDP. The difference with other signaling is that LDP operates on the basis of IP routing, instead of end-to-end signaling. Thus, the basic functions of MPLS can not be performed with the use of ICC of SCOMS software. The operation of an LDP-based MPLS control component is being standardized by IETF, and in SCOMS it is implemented separately from the other signaling software.

5.2.1 Enabling IP and MPLS in SCOMS environment

New IP-based hardware components include IP-routing capable Ethernet cards and MPLS capable ATM cards; thus, SCOMS switch designed to work as an edge router of an MPLS network. New software components include IP protocol suite, MPLS control component, LDP, IP routing and API extensions. Since the nature of SCOMS switch, interworking functions with other supporter network types are also considered.

Since the control software is run on a separate workstation and connected to the switch via ATM, the distinction between multilayer control and forwarding components is very clear. The solution has similarity with IP Switch that is also controlled by an Intel-based workstation. In fact, Ipsilon’s GSMP was used for controlling SCOMS switch before it became necessary to develop a specific API for the task. Unlike IP Switch, however, SCOMS switch is control-driven as only the control data is handled in the control component.

Multilayer switching requires running IP on control software, because IP routing protocols run on it. Running LDP requires also TCP and UDP. The

TCP/IP protocol suite of Linux is enabled with Classical IP over ATM (CLIP), that is a part of ATM on Linux distribution. It binds IP addresses to ATM VCs, and therefore makes it possible to transfer easily IP-based control data coming through MPLS and Ethernet interfaces to the control component via ATM. Each physical interface is given an IP address, which is also a requirement of the routing protocol. The physical interfaces of the switch are seen at the control workstation as logical ATM interfaces, bound to the IP layer. Consequently, the IP layer of Linux obscures the underlying ATM-based transfer path, and makes it possible to use normal socket operations on IP, TCP and UDP protocols, based on IP addresses and ports instead of ATM addresses and VCs.

The solution is also backed up by the MPLS specification for ATM-based networks, which requires having a non-MPLS connection between LDP peers, capable of transferring unlabelled IP packets. This non-MPLS connection is used for carrying LDP packets between two peers, and may be used for carrying other IP traffic, such as IP routing packets. [Dav2000b] Required Logical Link Control / Subnetwork Attachment Point (LLC/ SNAP) encapsulation is also available in the CLIP software. Data that is transferred between Ethernet interfaces and the control software has NULL encapsulation; IP packets are set straight to lower layer PDUs.

5.2.2 MPLS control component

Specification [Bos2000] introduces a set of common FSMs for processing LDP messages in an ATM-based LSR. These state machines are the center of the MPLS control component for the SCOMS switch. The specification proposes two sets of state machine tables that use downstream-on-demand mode, meaning that an upstream LSR asks downstream LSR to assign a label for a given FEC. One of these state machine sets can be used for non-VC-merge capable LSRs, while the other can be used for VC-merge capable LSRs. Merging several upstream labels to one downstream label is logical since IP routing does it automatically by forwarding packets from different sources with the same destination to one path. The specification proposes also a state machine set for downstream unsolicited mode, which means that a downstream LSR can distribute a label even if it has not been explicitly requested by an upstream LSR.

The state machine using downstream-on-demand mode and capable of VC-merge was chosen to be implemented, since it is recommended for ATM switches that have limited merging capabilities. It consists of three complete FSMs and one definition of general message handling that can be written to the form of a one-state FSM, "MPLS switch control" in Figure 9.

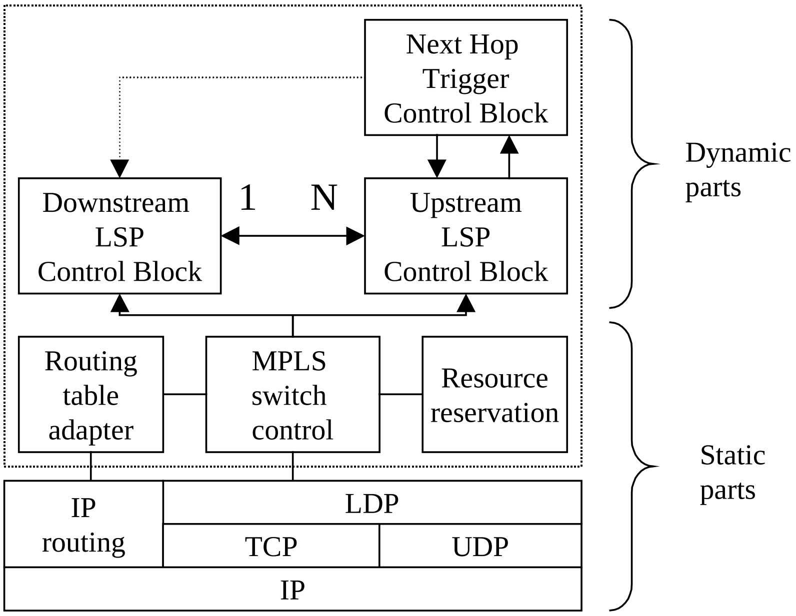


Figure 9. MPLS control component FSMs.

There is one Downstream LSP Control Block for controlling the downstream side of an LSP, and one or more Upstream LSP Control Blocks for controlling the upstream side. Next Hop Trigger Control Block is used when it is necessary to change an LSP for a better next hop, for example in the case of a routing change. Each control block contains sufficient information about the LSP, such as corresponding labels, FECs and LDP sessions. MPLS Switch Control FSM handles the messages arriving from

LDP, delivers messages from the control blocks to the correct LDP sessions and reacts to routing changes. It is also a central access point and storage for the blocks and it utilizes the reservation of resources as well.

It appeared that the FSMs could quite easily be converted to support both VC-merge and non-VC-merge capable switches. It is possible to freely configure the number of Upstream Control Blocks attached to a Downstream Control Block. Therefore, if the ratio is 1:1 no VC-merge is used.

Both ATM signaling software and the MPLS control component may be used for controlling the same physical transmission media, and the ATM virtual channel space is divided between the different technologies. Consequently, two separate logical networks are formed over the same physical network; one of which utilizes IP routing, and the other is based on ATM signaling.

5.2.3 IP and MPLS software architecture

Since the MPLS and IP solutions for the SCOMS switch are separate from the original signaling software, software development has been limited only by the restrictions of the protocol framework and the operating environment. Interfaces of different software modules are specified in interface classes to clarify the structure and make it easier to adopt possible changes or replace components that are specific to the hardware or operating system. Main software components concerning the new software architecture include (Figure 10):

- **API-FCF** that is used for controlling the switch. The FCF module is used for handling all internal event of the device, such as switching and the delivery of IP routing and MPLS information. By changing the FCF module other switches can be controlled with the software.

- **Routing table adapter** that reads and writes the routing table of Linux. The kernel of the operating system notifies about changes in the routing, and the adapter updates the hardware routing tables of the Ethernet interfaces and informs the MPLS control component of the changes. By changing the adapter, also other routing tables or routing protocol information can be used.
- **Label Distribution Protocol** contains a mechanism for opening and maintaining LDP sessions with LDP peers. It also encodes and decodes LDP messages between binary form and the message form of the framework.
- **Interworking functions** allow us to create interworking functionality between MPLS signaling and other networks specific signaling protocols. Interworking aspects are discussed in the next chapter.
- **MPLS control component FSMs**, as introduced in Figure 9. The core of the MPLS implementation uses LDP to open LSPs through the network. API-FCF is used for switching the LSPs in hardware and updating the forwarding tables. A routing table adapter is used for reading the IP routing table. Special interworking functions are not necessary inside the control component, since the interworking functions are connected to it with the same internal inputs that are used between the MPLS FSMs.
- **Configuration** is used for opening a correct set of LDP sessions, setting IP addresses of the interfaces and assigning IP addresses for special purposes. An IP address can, for example, be bound to a telephone number, thus allowing us to map signaling between IP and PSTN networks.
- **OSPF** is an essential part of the router functionality, but it is not integrated as part of the MPLS software architecture. Consequently, the solution is independent of the routing protocol. The OSPF implementation that is utilized in SCOMS has been implemented in the Calypso IP project of HUT [Cal2000].

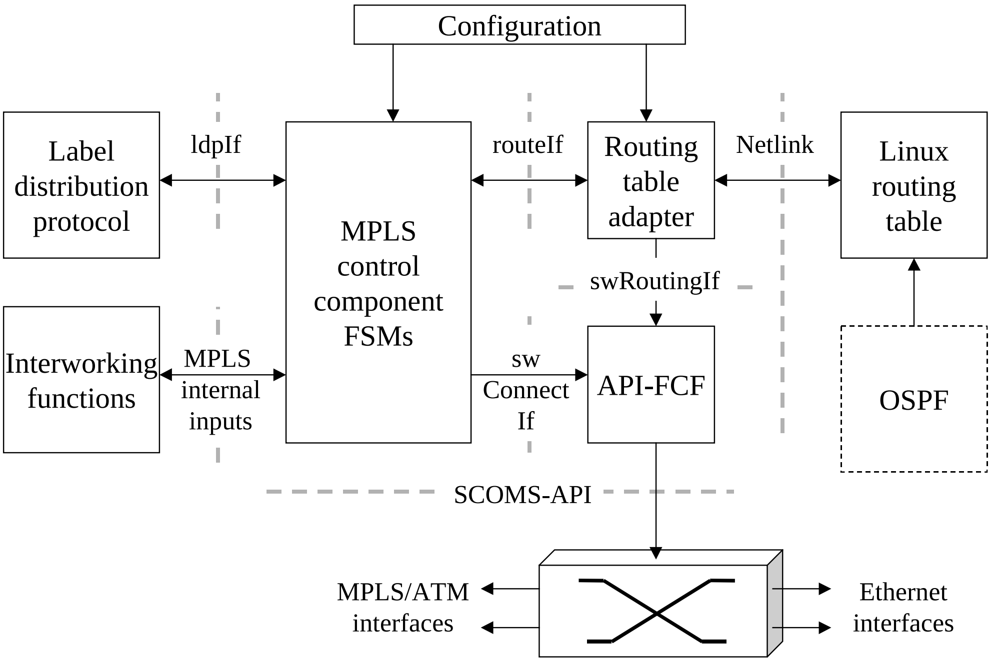


Figure 10. MPLS and IP routing software architecture.

5.2.4 MPLS interworking aspects

Since CR-LDP is a signaling protocol, capable of managing connections of individual application flows and other special services, it also allows interworking with other connection-oriented networks and protocols. CR-LDP is not specified as a separate FSM; instead it is implemented in the form of special information elements, interworking functions and interfaces to the MPLS control component. Connecting the MPLS control component to the ICC module by using a generic interface allows us to interconnect the MPLS network with any other network that is equipped with signaling functionality (Figure 11). Since the CC module can not be used for MPLS control, it is replaced by the MPLS control component. Each network interconnection case is implemented as a separate SCC protocol state. Using IP signaling, such as SIP as the other side of interworking would allow us to open all-IP connections with guaranteed QoS characteristics. [Sui2000b] CR-LSPs may be opened as bi-directional to save labels.

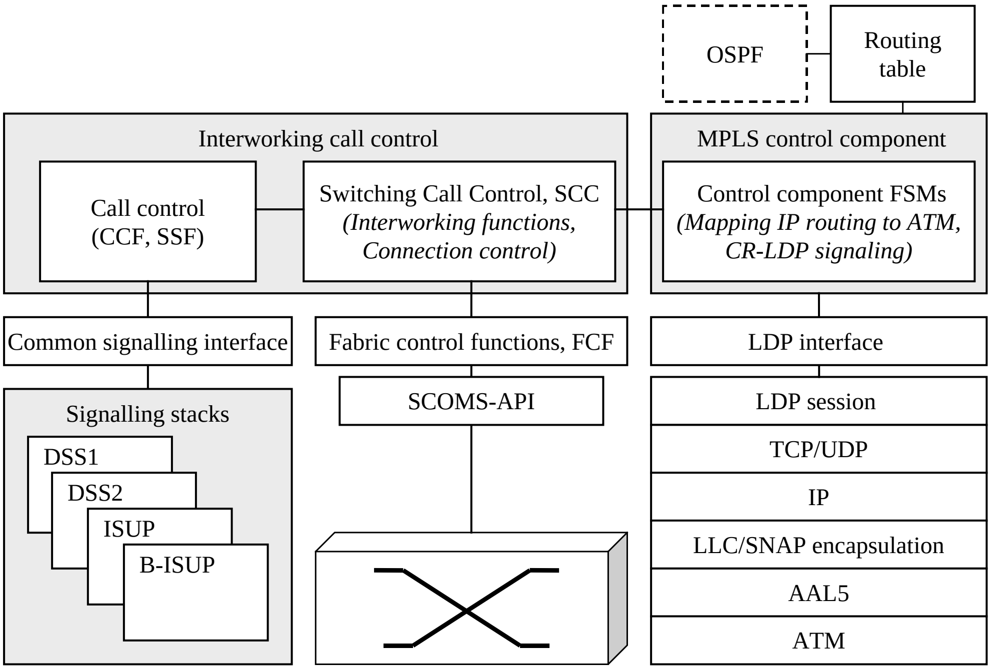


Figure 11. Interworking with MPLS [Sui2000b].

When a multipurpose switch is equipped with connection-oriented interfaces, other than those designed for transferring IP, additional consideration is needed to enable interworking with them. User data conversions are needed; for example, voice data is packed and usually encoded to achieve smaller data rate when it is transferred over an IP network. Therefore, interworking with an MPLS network is meaningful only if the user data is IP-based traffic or alternatively the data is converted from stream form to IP and vice versa on the hardware. Therefore, it is logical to route the streaming data from narrowband circuit networks through ATM virtual channels and IP-based data from IP access networks through MPLS LSPs, if additional data conversions are not performed [Sui2000b].

SCOMS software has routing capabilities used by signaling protocols and one issue is bringing it and IP routing together. Interworking with IP and signaling networks requires besides conversions between network addresses and telephone numbers, also the rational use of different routing schemes. Each case of interworking must be considered separately, but the basic approach is that in an interworking situation IP routing is used on the side of the IP network and the other side is controlled by the original routing of SCOMS software.

As noted before, sometimes it might not make sense to do interworking with an MPLS network. However, from the viewpoint of the MPLS control component the type of signaling protocol on the other side of SCC is nearly irrelevant, since the SCC hides the details of the other side. Only the type of routing on the terminating side must be known to initiate interworking. Thus, the same model applies, whether the other side is a PSTN, ISDN, ATM or IP signaling protocol. The following examples describe the two cases of interworking:

Ingress LSR

A connection request arrives at the incoming link (Figure 12). This results in the creation of a new signaling protocol instance and a new CC instance. In the case of an IP access network, CC could be replaced with an IP signaling protocol. CC reserves resources on the originating side, and uses routing to get the terminating side link identifier. Cross Connector Mux (CCM) is used to invoke the action of terminating side factory that would normally create an outgoing signaling protocol and a CC instance. In the case of MPLS a new Downstream LSP Control Block is created and connected to the originating CC via SCC. The incoming telephone number is replaced in SCC with an IP address that is obtained from a separate conversion table. SCC also maps messages between the originating CC and the Downstream LSP Control Block (appendix 1). Now the outgoing side reserves resources and sends a CR-LDP signaling message. Inside the MPLS network IP routing is used to open the path, and the LSP creation is handled by the MPLS control component only. Thus, interworking functions are not required again until the end of the LSP.

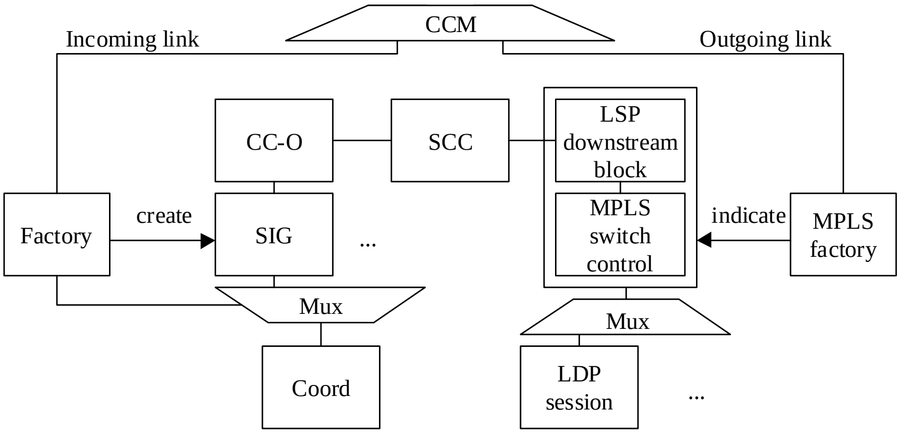


Figure 12. Interworking in an ingress LSR.

Egress LSR

An Upstream LSP Control Block receives a CR-LDP signaling message. IP routing is used for obtaining the next hop of the path. The next hop is identified as a non-IP type, so the routing scheme is changed. SCOMS routing gives the outgoing link identifier for the destination IP address, and the terminating side signaling protocol and CC are created with the use of CCM. The IP address is replaced in SCC with a telephone number and the terminating side continues to open the connection. SCC maps messages between the Upstream Control Block and terminating side CC (appendix 2).

5.2.5 Summary

Interworking requires a certain amount of configuration. Special purpose IP addresses must be set to the egress IP routing table and their corresponding telephone numbers configured to the conversion tables in both ingress and egress LSRs. Inside the MPLS network no configuration is required because the routing protocol distributes special IP addresses and calculates their shortest path routes.

The need for MPLS interworking can be justified with the emergence of new applications, such as Voice over MPLS (VoMPLS). In addition, the whole idea of SCOMS is to allow "any-to-any" interworking between the supported networking technologies. The inevitable result, however, is that all the different cases of interworking can not be backed up with standards, because they simply have not been standardized.

In Figure 13 a comparison between the IP/MPLS implementation and the SCOMS signaling software solution is illustrated. Note that the protocol stacks do not directly correspond to OSI layers. Instead, the picture illustrates relevant software components in a layered configuration and the individual stacks should be viewed separately. The only connection between IP/MPLS and the original signaling are the interworking functions on the top level and shared ATM resource reservation. Both ICC and IP/MPLS implementation use API-FCF to control the switch.

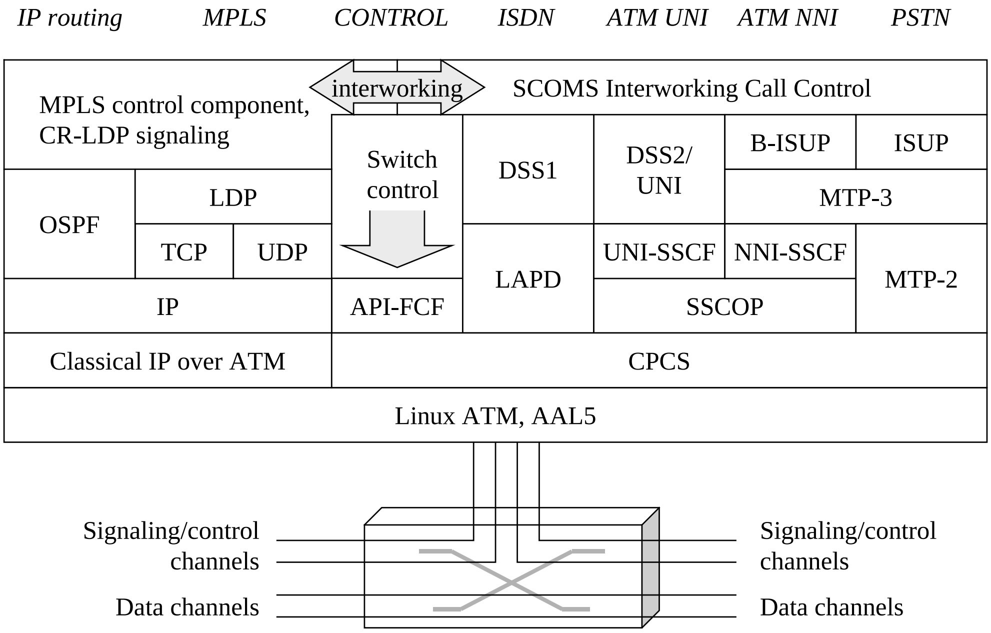


Figure 13. IP routing, MPLS and signaling stacks.

6 ANALYSIS

This chapter analyzes how well the solution fulfils the problem statement and how many of the criteria presented in chapter 3 are completed. The aim of this work has been to create an MPLS control component that uses LDP and IP routing, and to enable opening paths for individual calls inside an MPLS network. Many of the other features are currently incomplete at their best. They are seen as possible future enhancements, each of which needs a considerable amount of additional research. Thus, when a criterion of the "perfect" solution is not fulfilled by the current implementation, a simplified approach that is considered applicable at present is introduced as a solution to the problem. Some features require more discussion than others.

6.1 Functionality

Fine/coarse-grained QoS support for static and dynamic needs

As noted before, there are two types of IP QoS: Intserv and Diffserv. Intserv require resource reservation through the network for individual calls while Diffserv is based on traffic or service characterization and traffic prioritization.

Both static and dynamic support for Intserv is implemented. "Static" paths are based on IP routing, and while the routing is dynamic due to the use of a routing protocol, the resulting paths can be considered rather static than dynamic because they base on the network topology. Routing changes occur when some alterations in the network take place, and the changes are adapted to the switched paths as quickly as possible with LDP. These static paths guarantee only best effort QoS that is typical in the Internet.

If other QoS characteristics are needed, it is possible to open unidirectional or bi-directional CR-LSPs with QoS guarantees for single routes or individual application flows and calls. Signaling interworking is used for opening these connections and obtaining necessary traffic parameters. Thus, IP signaling as well as signaling of other networks may be used for access.

Supporting Diffserv would need additional IP packet header handling on the hardware, and additional functionality in the control software to map Diffserv Behavior Aggregate (BA) classes to LSPs. A BA consists of IP packets that cross the same link and require the same behavior. Specification [Fau2000a] introduces the operation and LDP extensions for Diffserv support, but converting the MPLS control component would probably require a considerable amount of work.

Routing and signaling protocols must also be extended to support Diffserv. Extensions to OSPF and CR-LDP (among others) are introduced in [Fau2000b]. They specify how the classes of Diffserv are carried in the network. In addition, some form of policy should be applied to control the use of Diff-Serv Code Points (DSCP) which are encoded in the headers of IP packets and identify their BAs. It is also important to be able to limit and monitor the use of high QoS resources.

One goal of MPLS is to offer quality characteristics that correspond to other modern IP technologies [Dav2000]. This means offering QoS based on both Diffserv and Intserv. The current implementation reaches this goal by half, since only the Intserv type of QoS is provided. Due to the problems presented above, supporting Diffserv is not a simple thing to do, requiring changes to major parts of the software as well as the hardware.

Interworking ability with other communication networks

The generic software solution allows us to do interworking with various networks, both IP-based and connection-oriented. Thus, one of the most important and interesting features of the whole SCOMS concept, making it possible to freely interconnect different networking technologies, is possible also with IP-based networks. However, a common IP signaling protocol should be deployed to allow us to open all-IP connections with guaranteed quality characteristics. This possibility is discussed further, and a concept of adapting SIP functionality to the platform is introduced in [Sui2000b].

Plain IP routing and the routing of SCOMS signaling software may not be enough to provide the variety of services that are possible with the SCOMS concept. SIP, for example, would need a separate location server that would be used in processing the SIP Uniform Resource Locators (URL) [Sui2000b]. The use of location server(s) and other centralized information bases might also be necessary to improve the mobility aspects of the solution. Mobility has not been considered in the project, but one possible use of SCOMS switch could be as part of a mobile network, offering gateway and network interconnection operations. In such an environment, traditional IP routing can not offer the dynamics needed for mobility.

External information bases could be accessed with Common Object Request Broker Architecture (CORBA). CORBA has already been used in SCOMS architecture for distributing separate software components such as the routing service. The use of CORBA in SCOMS has been examined in [Pär1999 and Num1999].

Traffic engineering facilities

Traffic engineering is concerned with performance optimization of operational networks. It does not include capacity expansions or traffic regulation but instead the efficient allocation of available resources. Performance objectives can be divided into two parts:

- **Traffic oriented** performance objectives include the aspects that enhance the QoS of traffic streams.

- **Resource oriented** performance objectives include the aspects aiming to the optimization of resource utilization [Awd1999b].

Minimizing congestion is the primary traffic and resource oriented performance objective. The efficient use of bandwidth resources helps to avoid a situation in which some parts of the network are congested while other parts are underutilized. One way to solve the traffic engineering problem is by using routing, since routing determines the paths taken by the traffic [Dav2000a].

Because best effort IP routing is not enough to solve the problem, some new approach must be taken. A logical way to enable traffic engineering would be extending the current routing protocol to support it. OSPF traffic engineering extensions are currently under work by IETF. Taking this approach would be convenient from the viewpoint of MPLS, since it does not require big modifications to the control component.

Traffic engineering with OSPF bases on the knowledge of available bandwidth inside the network. OSPF would have to be extended with QoS routing extensions, introduced in [Apo1999] and it should be able to receive the bandwidth information from somewhere, possibly from the MPLS control component. In that case, there would have to be a mechanism to exchange the information between MPLS resource reservation and OSPF.

Consider a case in which an ingress LSR receives a signaling message that requests a QoS path. Two potential ways for QoS path reservation and appropriate routing-based traffic engineering can be outlined:

- CR-LDP is used with hop-by-hop manner for opening a CR-LSP with certain QoS characteristics. The route is the same that best effort routing gives. On each LSR on the route, the MPLS control component informs OSPF about the reserved bandwidth and OSPF reacts to it by recalculating the routes, possibly making changes to other routes accordingly.

- The MPLS control component of the ingress LSR requests OSPF to calculate the best possible route with certain QoS requirements. When OSPF has calculated the route, it delivers an explicit route with all the intermediate nodes to the control component. CR-LDP is used to open an explicit CR-LSP through the network.

If a routing protocol independent approach must be taken, it is possible to solve the problem using only constraint-based routing to reach the objectives of traffic engineering. In this scenario, the MPLS control component would need several additions and modifications that are not defined in MPLS specifications. These include mechanisms for exchanging and maintaining topology state information, interactions between constraint-based routing and conventional routing, and mechanisms to accommodate adaptivity, resilience and survivability requirements of traffic trunks. [Awd1999b] It would seem that this approach is more complex and less applicable.

Virtual Private Network support

Nowadays the most common technique for providing a VPN service is with the use of IP over ATM overlay model. Each site in the VPN has a router that is connected via point-to-point links to routers in other sites. Since the ATM control is still available, we may continue using the old way for providing VPN services. The motivation for MPLS VPNs arises from the problems with the overlay networks. Scalability issues, high amount of configuration to be kept updated and the expertise in IP routing required from the customer are the major faults with the overlay model [Dav2000].

MPLS specifications do not specify a single and general way for constructing VPNs and the VPN architecture based on MPLS is still under discussion in ITU-T. Generally, the two basic approaches for MPLS VPNs are:

- Overloading some semantic(s) of existing routing protocols to carry reachability information.
- Virtual Routers (VR) that provide routing and forwarding services to the VPN customers [Mul2000].

The first approach has been taken in [Ros1999b] that introduces a method for providing VPN services with MPLS and BGP routing protocol. VPN routes are distributed via BGP and no other special protocols are needed. In SCOMS environment, however, we need to find a solution that bases on OSPF or preferably is independent of the routing protocol.

The second approach bases on the VR concept, which has the same mechanisms as a physical router and therefore inherits all existing mechanisms and tools for configuration, operation, accounting and maintenance. Within a VPN domain, an instance of routing is used to distribute VPN reachability information among VR routers. In principle, any routing protocol can be used, and no VPN-related modifications or extensions are needed for the routing protocol to achieve VPN reachability [Oul2000].

A VPN customer site is connected to the provider backbone with a connection between the Customer Equipment (CE) device, (which can be either a bridge or a router) and the VR that is located in the Provider Edge Router (PE). Virtual routers have independent IP routing and forwarding tables and they are isolated from each other. A VPN may use whatever addressing is needed. This unfortunately requires modifications to the current OSPF implementation, since the same routing table that is used for IP routing can not be used for the VPN. Other possible modifications may arise from the fact that routing instances need to be separated for VR purposes. The details of the routing protocol, however, are out of the scope of this study.

A VR must also be able to forward packets to the next hops within the VPN domain. We might do the forwarding in the control workstation, which would require creating a forwarding engine that uses the separate VR routing tables. This approach would need a considerable amount of extra work, and the idea is against the control-driven nature of the SCOMS switch. If the forwarding were done on hardware, implementation would depend on the type of access network. Since we are discussing IP-based VPN services over an MPLS network, it may be noted that connection-oriented access interface would be based on configuration of point-to-multipoint layer 2 connections and no IP routing would be used for forwarding. Using the IP forwarding engine of an Ethernet card would mean that the card in question is connected to a VPN domain and actually corresponds to a CE device that is controlled by a VR. Therefore, the hardware routing table would indeed be completely separate from the IP routing of the LSR and no changes to the forwarding engine would be needed. The control software would need to be able to maintain separate VPN routing tables, which would be delivered to VPN interfaces and an LSR routing table that would be delivered to the interfaces not connected to the VPN domains.

The connections inside the MPLS domain would be tunnels, which means that the IP headers of packets delivered inside a VPN domain are not associated to the IP routing of the MPLS network. The virtual router approach explicitly separates the mechanisms used for distributing reachability information from mechanisms used for achieving VPN topology determination [Oul2000]. VPN topology discovery could be done with the use of a directory server, which VRs would query to determine their neighbors. The discovery of a VR belonging to the same VPN would trigger a CR-LSP creation between the VRs and VC-merge would be used inside the MPLS network where possible. This would be a considerable improvement over configuration-based and static ATM VPN services. Naturally, the MPLS VPN connections might also be based on an explicit configuration.

Multicasting capability

Multicasting may be used for sending data to multiple hosts over the network for purposes of audio- and videoconferencing, multimedia delivery and other services that require one-to-many communications. Users join a multicast group using Internet Group Management Protocol (IGMP). A multicast group is identified with a certain group address, and multicast routers of the network handle the data delivery between the participants. A multicast routing protocol is needed to calculate the least cost paths between the packet’s source and any particular destination group member. In this context the use of Multicast OSPF (MOSPF) [Moy1994] is considered. A multicast packet’s path is calculated by building a pruned shortest-path tree rooted at the packet’s source. Routing with MOSPF differs from that of unicast since a multicast packet is routed based on both the packet’s source and its multicast destination.

The principle of possible multicasting implementation is as follows: at ingress LSR is the source of one multicast tree. MOSPF calculates the routes of the shortest-path tree which is mapped to LSPs. Members can be added or removed dynamically. In the network, there is one multicast tree per each member of a group. This results from the fact that MOSPF does not support trees that are shared with different source addresses, and multicast routing protocols in general do not support the aggregation of multicast trees with different destination addresses. [Oom2000]

To support multicasting, modifications are definitely needed to the MPLS control component. LDP support for multicast is currently left for future study by IETF, and the control component specification does not discuss multicast possibilities either. The framework for MPLS multicasting, proposes the following choices for multicast LSP creation:

- Request-driven: intercept the sending or receiving of control messages and use the information to bind LSPs to routes
- Topology-driven: obtain the multicast tree from a Multicast Routing Table (MRT) and map it to a level 2 tree
- Traffic-driven: map the tree when multicast data arrives [Oom2000]

Request-driven approach requires intercepting and handling MOSPF messages, resulting in the processing of the same messages twice. Traffic-driven approach requires additional functionality to hardware to be able to inform the control software about arriving multicast messages, as well as to buffer the multicast packets while LSPs are being created. Topology-driven approach appears to be the most applicable.

Each MOSPF router in the path of a multicast packet bases its forwarding decision on the contents of a forwarding cache. There is a separate forwarding cache entry for each source/destination combination. A forwarding cache entry is built from two parts. The first is called the local group database. This database, built by the IGMP protocol, indicates the group membership of the router’s directly attached networks. The second component is the packet’s shortest path tree [Moy1994]. Thus, it is possible to obtain the multicast tree at every LSR in the multicast path, which increases possibilities for the implementation of LSP creation. An LSR always knows both the next hop and the previous hop, because the multicast tree is constructed using the source/destination routing. Four primary problems concerning the implementation exist:

- The MPLS control component supports VC-merge for a given destination address, which is analogous to multipoint-to-point ATM connections. Multicasting requires VC-merge for a given source address, analogous to point-to-multipoint connections. That is, if we do not wish to maintain end-to-end paths from each source to every destination. This would spoil the idea of multicasting by unnecessarily wasting bandwidth and labels.

- Multicast data may be forwarded to several outgoing interfaces, not all of which are necessarily of the same type. For example, an arriving multicast packet might be replicated to two packets (if using VC-merge on the incoming link), one of which is forwarded to an Ethernet interface, and another to an MPLS interface. Multipoint connections with different types of destination interfaces have not been considered in the implementation.
- Time To Live (TTL) field of IP headers is used for limiting the multicast area. Inside an ATM-based network, IP headers are not processed and thus there is no way to decrement the TTL value.
- MOSPF creates the shortest-path trees on demand; they are created when the first multicast packet is received. How is it possible to make use of a traffic-driven routing protocol when the MPLS network is control-driven?

Making arbitrary multicast extensions to the standard-based control component might not be a great idea, but there are two possible approaches for extending the current control component to be multicast-capable. Assume that there is a mechanism for the MPLS control component and MOSPF to exchange multicast information.

- Source-initiated tree creation. A special multicast mode is implemented to the existing (Figure 9) FSMs. In this mode, an Upstream LSP Control Block is able to store several pointers of Downstream LSP Control Blocks. Each FSM is modified to take action depending on the mode, and there is a strict distinction between unicast and multicast modes. Perhaps even a special FSM such as "Multicast LSP Upstream Block" should be implemented to support multicasting. This solution would require considerable changes to the control component, but if the multicast implementation was kept separately from that of the unicast, it would not interfere with the normal MPLS operation.

- Destination-initiated tree creation. Unfortunately, Downstream-on-Demand mode does not allow a downstream LSR to open LSPs without a request from upstream and therefore the control component does not offer functionality to it either. It is clear now that Unsolicited Downstream mode would have solved this problem and allowed opening an LSP from each multicast destination towards every source. However, the merge capability would still have had to be implemented to control blocks. A possibility with the current implementation is to use a destination-initiated LSP creation and reversed transport of data. In this scenario, when an LSR finds out that it is a destination for a given source, it initiates a CR-LSP creation towards the source. Thus, a tree that corresponds exactly to a multicast tree is created with a normal merge-capable operation of control component with only one exception. Instead of using the LSPs to transport the data towards the multicast source address, they are used in the reverse direction. This must be enabled at the root of the tree where a multicast address is bound. At each destination, a layer 3 routing operation is performed and the multicast packets are sent to the clients.

The problem with different types of destination interfaces is not a difficult control problem. If we are able to connect an incoming LSP to several outgoing connections or interfaces (which is not currently possible), each connection is created separately. Thus, we have the information about the types of interconnections available to be processed and stored. How the switch can handle the multipoint connections with varying interfaces is a hardware problem.

The TTL problem is a difficult one. With a unicast LSP the TTL field could be decremented at the ingress or the egress LSR. Multicast branches, however, can have different lengths so the TTL can only be decremented at the egress LSR. This potentially wastes bandwidth if the TTL turns out to be zero or negative [Oom2000].

If MOSPF extensions were really made to work in a traffic-driven way, the protocol would not be applicable without traffic-driven modifications to MPLS control component and hardware. A better approach would be to calculate multicast trees always when someone joins a multicast group. Naturally, this could result in wasted resources if the trees were mapped to an MPLS network even if no multicast data was transferred.

MPLS multicasting requires solving several issues. One big problem is that instead of one superior multicast routing protocol there are several equal ones that may still co-exist in the future. An interoperable solution can not be created before IETF decides which is the best way to enable multicasting.

6.2 Applicability

Scalability

ATM technology has limitations that affect scalability. Virtual channel identifiers limit the amount of labels available, and then there is the question of limited bandwidth. In principle, there may be an arbitrary number of ATM cards in the switch, each of which are capable of having a theoretical maximum of 2^{28} connections. The number of connections is limited by the length of the VPI and VCI fields (12 + 16 bits), since the label space is reserved on per-interface basis. A more important concern is the current distributed control software solution. Processing power and the amount of memory in the control workstation are limiting factors and object-oriented methods used in control software development do not consider performance. The problem is how well the handling of control data scales in the network, and how the amount of it could be minimized.

VC-merge is an efficient way to save the limited label space, when performing the basic MPLS operation of mapping IP routing to the ATM level. In a network with n sources and destinations, a non-VC-merge capable LSR has to potentially manage $O(n^2)$ VC labels for full-meshed connectivity. With VC-merge, an LSR is required to manage only a minimum of $O(n)$ VC labels. [Wid1999] The current implementation does exactly this. A possibility for even more efficient label use would be utilizing some policy to configure the aggregation of different routes ending up to an egress LSR, an approach taken by IBM ARIS. This would require changes to the control component, but since the merging of labels is already done, it might not be very difficult to aggregate also FECs if sufficient policy information were available.

Memory requirements may be calculated as static and dynamic requirements. Figure 9 illustrates the static and dynamic parts of the MPLS software. Creating one LSP inside an MPLS network requires a Downstream LSP Control Block and an Upstream LSP Control Block. With VC-merge, there is one Downstream LSP Control Block per FEC and as many Upstream LSP Control Blocks as there are upstream LDP peers.

The states in FSMs are implemented using Singleton design pattern, thus the state instances are common for all block instances and belong to static memory requirements. Running the static parts needed for MPLS operation requires about 2 megabytes of memory. When testing with 100000 LSPs (both Upstream and Downstream LSP Control Blocks were created) memory consumption grew by 59 megabytes. An IP network with more than 100000 sources and destinations would have to be very large and other constraints such as processing power, manageability or bandwidth would become significant barriers. Therefore, it may be concluded that the amount of memory is not an issue if we support VC-merge when making connections.

The processing power of the control workstation is significant when discussing the creation of dynamic connections. The scheduler of the OVOPS++ framework share processing time for protocol functions, and signaling may be delayed if a great number of events are waiting for computing. While this is a problem with connection-oriented networks, MPLS is not so greatly affected because of the routing-based and control-driven action. When a new route is discovered in the network, LSP creation starts immediately and a switched path is established presumably well before the first packet of user data arrives. CR-LDP corresponds to other signaling protocols and is more affected. The implementation uses "ordered" control mode of label distribution; the delivery of user data at the ingress LSR is not enabled before an end-to-end connection is ready. If a user data packet arrives before a connection is available, it is rejected. A broad test environment would be needed to be able to measure the performance of the control software. An interesting subject of research would be the performance of an object-oriented protocol framework, such as OVOPS++.

Opening LSPs for individual calls or application flows consumes labels as well as the precious bandwidth. A fundamental question concerning scalability is, whether it is even possible to construct a scalable network that offers Intserv type of QoS. The question is not an easy one, but it is obvious that an efficient admission control and a billing mechanism would be needed to control the amount of users that require guaranteed QoS paths. Diffserv type of QoS would be a more scalable way, since it is based on traffic or service characterization instead of maintaining per-flow reservation through the network. It would be very useful to be able to aggregate several fine flows into one coarse flow.

It is also questionable whether an Intel-based control workstation running Linux has performance high enough to control operations in a large network. A commercial product might need a multiprocessor computer with enough redundancy, controlled use of resources and failsafe mechanisms.

Independence of the routing protocol

Since there is no need to obtain routing or link state information directly from the routing protocol, the information is read from the routing table of the operating system. Each change in the routing table is handled and considered separately, and the procedure is the same regardless of which routing protocol or configuration event changes the table. This applies to the current IP/MPLS implementation.

As it has been shown, certain applications such as VPNs, multicasting, traffic engineering and Diffserv could require binding oneself to a specific routing protocol. It is difficult to maintain the independence of the routing protocol if some routing related data that is not available through the Linux kernel interface is being utilized. A good approach would be to store relevant information in a Management Information Base (MIB) and modify each protocol or other part of the software to be used with the MIB. The software framework, however, promotes a protocol like approach; each software component must be connected to others in a more or less layered configuration, and trigger necessary actions with messages sent through the protocol layers. Some trigger mechanism would have to be implemented in the MIB to achieve a similar operation.

Generality with respect to link layer technologies

MPLS technology is general, and can be used for any of the link layer technologies, such as Frame Relay, Ethernet, WDM or PPP (Point-to-Point Protocol). The implementation, on the other hand, is based on the FSMs that are developed for ATM switch LSRs. The specification does not contain anything strictly ATM-specific, so the implementation could easily be adapted for other types of LSRs as well. The modular software architecture would help to use different resource reservation, which is the most ATM-specific part, and thus make it easier to adopt MPLS to other than ATM technologies. Some minor changes would be required to the state machines, such as the ability to maintain different types of labels and API extensions to support different MPLS link layer technologies.

Portability and readiness for future updates

OVOPS++ has been ported to several UNIX operating systems such as FreeBSD, BeOS, Cygwin and IRIX, and thus the SCOMS software can be ported to them as well. Some Linux-specific solutions have been made with MPLS, such as the usage of CLIP. Other operating systems may or may not have their own procedures for creating corresponding IP over ATM connections. In principle, it is easy to change the Linux specific parts of the software if the same functionality is available in the other OS. A routing table adapter is another Linux specific part, since it uses a kernel interface to obtain routing information. The adapter has a common interface with other software modules; thus, it can be replaced with any adapter that is suitable for some other operating system.

API-FCF is the only part that is specific to the SCOMS hardware, and it can be replaced with an FCF module to allow operability with another switch. Since the SCOMS concept is unique, with other switching hardware only portions of the software can be used effectively.

Running IP version 6 network should not be radically different from IP version 4. Naturally, changes in address handling are needed, but the basic concept of binding IP routing to switched paths is still the same and major parts of the software remain unchanged.

6.3 Controllability

Operations, administration and maintenance facilities

Currently most of the OAM capabilities available are based on static configuration files. To enable more flexible OAM facilities it should be possible to configure all the information using Simple Network Management Protocol (SNMP). In addition, a MIB should be implemented to store the configuration information.

In SCOMS project, an Interim Local Management Interface (ILMI) has been implemented to support exchanging ATM management information. ILMI agent handles switch attributes using a CORBA interface, and decodes SNMP messages that are delivered inside the ATM network [Pär1999]. To be able to deliver SNMP messages over IP, an additional SNMP agent must be implemented.

MPLS specifications define currently three portions for the MIB:

- Common MPLS LSR information, which allows configuring interfaces, LSP segments, LSP connections, label stacks and traffic parameters [Sri2000a].
- Label Distribution Protocol information, which provides objects to configure/set-up potential LDP sessions, store information about LDP peers and actual established LDP sessions [Cuc2000].
- Traffic engineering information, which allows configuration of point-to-point unidirectional traffic engineering tunnels and their parameters [Sri2000b].

At least the first two portions would be useful for SCOMS purposes except that stacking labels has not been considered in the current implementation. MIB specifications do not currently allow some applications discussed in this chapter, such as VPNs and multicasting.

Minimal complexity, modular architecture

The software architecture has some complex parts, such as the core of the MPLS control component, that contains four different FSMs. It can be seen as one integrated software module, since it has not been forced to a layered configuration but to one that is more logical and easier to implement. Other logical modules such as LDP, OSPF and the routing table adapter are separated with clear interfaces.

Modularity is especially important in environment specific parts of the software, since they define how well the software adapts to other environments. Those parts of the SCOMS software solution that use operating system or hardware functions are in separate modules that can be easily replaced.

Obviously, the more of the improvements discussed in this chapter are implemented to the control software, the more complex the software architecture becomes. Adapting the same basic implementation for everything possible might make it impossible to maintain any logical or clear software structure and would therefore make the solution more error prone and difficult to maintain. On the other hand, keeping each software module strictly separate from others could easily cause duplicate code and repeating of the same features in several modules.

6.4 Performance

Control information handling does not delay user data

The MPLS implementation in SCOMS is control-driven, which means that a path through the network is opened before the first packet of user data traversing that path arrives. If a route is available, a switched path through the MPLS network is also established.

Control-driven operation ensures that the ATM technology is used at maximum speed and the control component does not cause delay to the delivery of the user data. In the case of opening CR-LSPs for individual calls, normal delay caused by an end-to-end signaling operation occurs and the performance of the control component is more significant.

6.5 Summary

Table 2 contains an evaluation of how well the current implementation fulfills the criteria set to the "perfect" solution of the problem statement. Status is on the scale from a minimum of 0 to a maximum of 5.

Table 2. Level of implementation of the criteria.

Criterion	**Status**
Fine/coarse-grained Qos support for static and dynamic needs	3
Interworking ability with other communication networks	4
Traffic engineering facilities	1
Virtual Private network support	1
Multicasting capability	0
Scalability	3
Independence of the routing protocol	5
Generality with respect to link layer technologies	3
Portability and readiness for future updates	4
Operations, Administration and maintenance facilities	2
Minimal complexity, modular architecture	4
Control information handling does not delay user data	4

7 CONCLUSIONS

The demand for better service quality in IP networks is wide and growing all the time, since almost all data delivery will soon be IP-based. Bandwidth-hungry applications and services, such as video and audio streaming, are becoming increasingly popular. These services need constant QoS characteristics in order to work properly, at least when using real-time services that do not allow large buffering of data.

At present, it seems that universal IP QoS will not be available for some time. The QoS problem can be solved technically at least to some extent, but more difficult financial and political problems exist. Bandwidth will be a limited resource also in the future, and for a network operator there are no motives for offering QoS paths through a public network if the users are not willing to pay for them. The Internet has always been free in the sense that users only pays for access and not for data delivery. Therefore, there is not going to be any universal billing mechanism either anytime soon. One practical way of offering QoS is through VPNs, because then the network operator can control the usage of bandwidth and most importantly knows whom to send the bill to.

MPLS has become an important networking technology, because it creates a bridge between IP and connection-oriented networks. It is also a very versatile technology that allows the network operator to deploy several useful and valuable services such as VPNs, IP QoS and traffic engineering. The unfortunate side effect of the versatility is that the once so simple and nice idea has swelled during the standardization to a monster that is almost incomprehensible in all its complexity and multiformity. The problem is common in all bigger networking standardization efforts. Every big company tries to get their solution as a part of the standard and particularly interesting technologies such as MPLS cause the emergence of a great number of specifications. Consequently, there is a basic set of features that almost everyone supports, such as LDP and CR-LDP. Then there are the numerous vendor specific additions that will never be implemented in all products although they are published as specifications. Naturally, this decreases the level of interoperability between the products of different vendors. MPLS products are currently being manufactured and marketed, but the industry is already looking into a new generation of MPLS equipment that will be most likely based on WDM technology instead of ATM.

In this work, I had to find an appropriate set of specifications to be able to implement a standard-based MPLS control component. Most of the available Internet drafts were skimmed through and then simply ignored. The resulting implementation is based on the specifications that were considered as most fundamental and that should allow basic interoperability with commercial products. However, we would not be making research if we did not try to create also something new. The most interesting SCOMS-specific feature is the network interoperability that allows interconnecting of both IP-based access networks and narrowband connection-oriented networks to an IP-centric MPLS network.

The result of the work is a software architecture with implemented modules, except those that are hardware specific since the IP extensions of SCOMS-API have not been completed at this point in time. Different interworking functions must also be considered one at a time and implemented separately for required interconnections. Quite a few elements have an effect to the overall functionality of the solution. Such elements include the MPLS control component, OSPF, LDP, handling of the routing table, ATM on Linux software, interworking functions, resource reservation, SCOMS-API, OVOPS++ framework, configuration and several other smaller parts of the software, as well as the new hardware. It should be obvious that a very extensive testing effort is required to make sure that everything works together. However, the structural solution is ready and can be done fully operational with sufficient testing.

The future will probably bring more solutions that allow the integration of different networks. Even if narrowband landline technologies are old and have their limitations, they provide a reliable transfer path and reach a large user base. Future mobile networks that are currently under development will hardly provide significantly greater bandwidth for people living outside cities. Therefore, it may be envisaged that landline telephone networks will be used for network access for a long time. It is also easier to provide broadband services through fixed lines, which will extend their lifespan even more.

Possible future enhancement for the SCOMS concept could be some new broadband user access interface or implementation of some new MPLS features that have been proposed in this work. One vision of the future is introducing IP-based service and load balancing capabilities, which would allow offering sophisticated IP-based services through heterogeneous networks.

True network convergence requires besides a common data transfer protocol also new and innovative solutions that are capable of handling the control mechanisms of different networks. Optimistically thinking, in the future a user could have a common user interface for all network services but not have to think about the actual data transfer path. The time of "any-to-any" networking is not anywhere near but we are getting closer.

REFERENCES

[3GP1999] 3rd Generation Partnership Project. Technical Specification Group Services and Systems Aspects; Architecture for an All IP network. 1999.

[Amo1998] Amoss, J. Minoli, D. IP Applications with ATM. McGraw-Hill. 1998. ISBN 0-07-042312-1.

[And2000] Anderson, L. et al. LDP Specification. Work in progress. IETF. August, 2000.

[Anr2000] Anritsu Corporation. The Must-Have Reference for Multilayer Switching. 2000. [Cited 1.7.2000]
Available: http://www.global.anritsu.com/products/infosys/musthave.html

[Apo1999] Apostolopoulos, G. et al. QoS Routing Mechanisms and OSPF Extensions. Request for Comments: 2676. IETF. August, 1999.

[Awd1999a] Awduche, D. MPLS and Traffic Engineering in IP Networks. IEEE Communications Magazine. December, 1999. pp 42-47.

[Awd1999b] Awduche, D. et al. Requirements for Traffic Engineering Over MPLS. Request for Comments: 2702. IETF. September, 1999.

[Bel2000] Understanding digital circuit switching. Bell Labs, Lucent Technologies. [Cited 2.6.2000]
Available: http://www.bell-labs.com/technology/circuit/

[Bos2000] Boscher, C. et al. LDP State Machine. Work in progress. IETF. January, 2000.

[Cal2000] Calypso IP project. Helsinki University of Technology. [Cited: 17.10.2000]
Available: http://www.tml.hut.fi/Tutkimus/G4/cal_ip.html

[Cuc2000] Cucchiara, J. et al. Definitions of Managed Objects for the Multiprotocol Label Switching, Label Distribution Protocol (LDP). Work in progress. IETF. August, 2000.

[Dav2000a] Davie, B. Rekhter, Y. MPLS: Technology and Applications. Morgan Kaufmann Publishers. 2000. ISBN: 1558606564.

[Dav2000b] Davie, B. et al. MPLS using LDP and ATM VC Switching. Work in progress. IETF. June, 2000.

[Fau2000a] Le Faucheur, F. et al. MPLS Support of Differentiated Services. Work in progress. IETF. August, 2000.

[Fau2000b] Le Faucheur, F. et al. Extensions to IS-IS, OSPF, RSVP and CR-LDP for support of Diff-Serv-aware MPLS Traffic Engineering. Work in progress. IETF. July, 2000.

[Fel1997] Feldman, N. Viswanathan, A. ARIS Specification. Work in progress. IETF. 1997.

[Fry2000] Fryer, J. IP + ATM = MPLS. Telecommunications Magazine. February 2000. [Cited: 5.6.2000]
Available: http://www.telecoms-mag.com/issues/200001/tcs/ip.html

[Gro1999] Grossman, D. Heinänen, J. Multiprotocol Encapsulation over ATM Adaptation Layer 5. Request for Comments: 2684. IETF. September, 1999.

[Gru1998] Grundström, M. Mickos, R. ATM-tekniikka ja monipalveluverkot, 2. Painos. Suomen ATK-kustannus Oy. 1998. ISBN 951-762-488-3. [In Finnish]

[Hal1996] Halsall, F. Data Communications, Computer Networks and Open Systems. Addison-Wesley Publishing Company. 1996. ISBN 0-201-42293-X.

[Hün1995] Hüni, H. Johnson, R. Engel, R. A Framework for Network Protocol Software. GLUE Software Engineering, University of Illinois, Ascom Tech AG. 1995.

[IpA1999] Transport of IP over ATM in public networks. Draft Recommendation I.ipatm. ITU-T. 1999.

[Jam2000] Jamoussi, B. et al. Constraint-Based LSP Setup using LDP. Work in progress. IETF. July, 2000.

[Lau1998] Laubach, M. and Halpern, J. Classical IP and ARP over ATM. Request for Comments: 2225. IETF. April, 1998.

[Mar1997] Marin, G. and Boivie, R. Digging into IBM’s ARIS label switching protocol. Network World Magazine. March, 1997.

[Moy1994] Moy, J. Multicast Extensions to OSPF. Request for Comments: 1584. IETF. 1994.

[Mpl2000] Multiprotocol Label Switching Charter. [Cited: 25.7.2000] Available: http://www.ietf.org/html.charters/mpls-charter.html

[Mul2000] Muthukrishnan, K. and Malis, A. A Core MPLS IP VPN Architecture. Work in progress. IETF. June, 2000.

[New1997] Newman, P. IP Switching – ATM under IP. IEEE/ACM Transactions on Networking, vol. 6 no. 2. 1997.

[Num1999] Nummisalo, P. Uudet olioteknologiat älyverkko-toteutuksessa. Diplomityö. Lappeenrannan teknillinen korkeakoulu. 1999. [In Finnish]

[Oom2000] Ooms, D. et al. Framework for IP Multicast in MPLS. Work in progress. IETF. May, 2000.

[Oul2000] Ould-Brahim, H. et al. Network based IP VPN Architecture using Virtual Routers. Work in progress. IETF. July, 2000.

[Per1995] Perez, M. et al. ATM Signaling Support for IP over ATM. Request for Comments: 1755. IETF. February, 1995.

[Pet1996] Peterson, L. Computer Networks: A Systems Approach. Morgan and Kaufman, San Mateo, CA, 1996. ISBN 1-55860-368-9.

[Pre1997] Pressman, R. Software engineering: a practitioner’s approach. McGraw-Hill. 1997. ISBN 0-071-14603-2.

[Pry1995] de Prycker, Martin. Asynchronous Transfer Mode: Solution for Broadband ISDN. Prentice Hall International. 1995. ISBN 0-13-342171-6.

[Pär1999] Pärnänen, T. Oliosuuntautunut ATM-kytkimen ohjaus-ohjelmisto. Diplomityö. Lappeenrannan teknillinen korkeakoulu. 1999. [In Finnish]

[Raa1999a] Raatikainen, S. Merkinanto ja puhelunohjaus monikäyttökytkimessä. Diplomityö. Lappeenrannan teknillinen korkeakoulu. 1999. [In Finnish]

[Raa1999b] Raatikainen, K. P. T. Multidiscipline Switching for Delivering Multimedia Services. Proceedings of ICT'99. Cheju (Korea). June 1999. Vol. 2. pp. 86-90.

[Raa1999c] Raatikainen, K. P. T. Interworking support in a multidiscipline switch. Proceedings of the the International Conference on Broadband Communications (BC'99). IFIP TC6.2. Hong Kong. 10-12 November 1999. Kluwer Academic Publishers, Boston. pp 395-404.

[Raa2000] Raatikainen, P. Raatikainen, S. An interworking call control solution for a multidiscipline switch. Proceedings of Networking 2000. Paris. 22-25 May 2000. Springer-Verlag. Berlin. pp 98-107.

[Rek1997a] Rekhter, Y. et al. Cisco Systems' Tag Switching Architecture Overview. Request for Comments: 2105. 1997

[Rek1997b] Rekhter, Y. et al. Tag Switching Architecture – Overview. Work in progress. IETF. 1997

[Ros1999a] Rosen, E. Viswanathan, A. Callon, R. Multiprotocol Label Switching Architecture. Work in progress. IETF. July, 1999.

[Ros1999b] Rosen, E. and Rekhter, Y. BGP/MPLS VPNs. Request for Comments: 2547. IETF. 1999.

[RHK1999] Press release: RHK Releases New Report on the Emerging Architectures in an IP World. [Cited: 5.6.2000]
Available: http://www.rhk.com/pressrelease.asp?id=28

[Sco2000] SCOMS project. Helsinki University of Technology.
[Cited: 17.10.2000]
Available: http://www.tml.hut.fi/Tutkimus/G4/scoms.html

[Sem1999] Semeria, C. Multiprotocol Label Switching. White paper. Juniper Networks. 1999.

[She1994] Shenker, S. Fundamental Design Issues for the Future Internet. Xerox Palo alto Research Center. 1994.

[Sri2000a] Srinivasan, C. et al. MPLS Label Switch Router Management Information Base Using SMIv2. Work in progress. IETF. July, 2000.

[Sri2000b] Srinivasan, C. et al. MPLS Traffic Engineering Management Information Base Using SMIv2. Work in progress. IETF. July, 2000.

[Sta1999] Stardust.com. The Need for QoS. White paper. 1999.
[Cited 12.6.2000]
Available: http://www.winsock2.com/qos/whitepapers/
need.htm

[Ste1994] Stevens, R. TCP/IP Illustrated, Volume 1. Addison-Wesley. 1994. ISBN 0-201-63346-9.

[Sui2000a] Suihkonen, O. Using OVOPS++ protocol framework. Helsinki University of Technology. 2000.

[Sui2000b] Suihkonen, O. Raatikainen P. and Raatikainen, S. A Signalling Interworking Solution for Heterogenous Networking. Helsinki University of Technology, Technical Research Centre of Finland. 2000. Accepted to IASTED International Conference on Applied Informatics 2001.

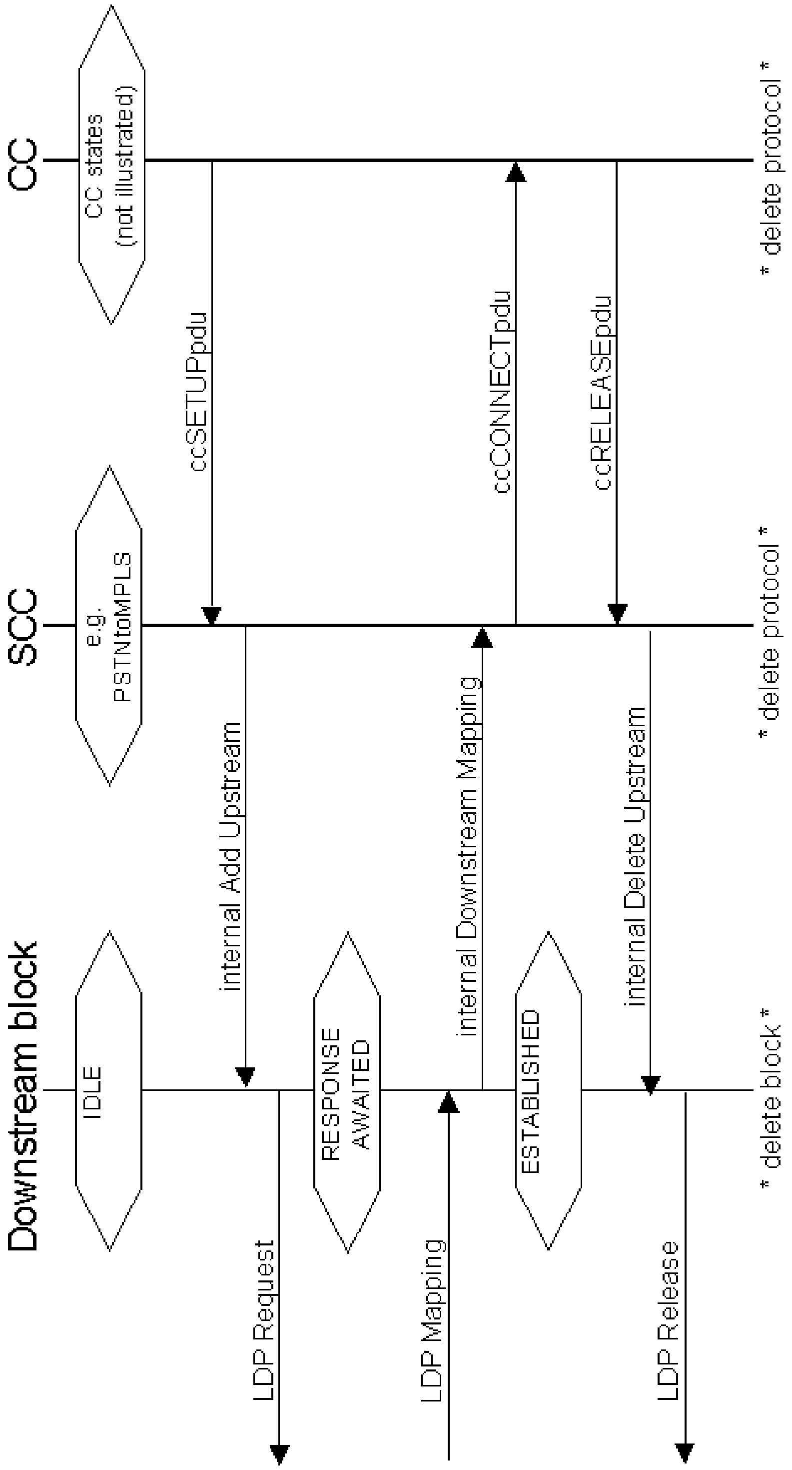
[Vis1997] Viswanathan, A. Feldman, N. Boivie, N. Woundy R. ARIS: Aggregate Route-Based IP Switching. Work in progress. IETF. 1997.

[Wha2000] whatis.com [Cited 2.6.2000]
Available: http://www.whatis.com

[Wid1999] Widjaja, I. et al. Performance Issues in VC-Merge Capable ATM LSRs. Request For Comments: 2682. IETF. 1999.

[Wit1997] Wittman, Art. IP Switching: Battle For The Network High Ground. Network Computing Magazine. June, 1997.
Available: http://www.networkcomputing.com/811/
811f1.html

APPENDIX 1. Mapping between CC and LDP at ingress LSR



APPENDIX 2. Mapping between CC and LDP at egress LSR