UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Provisioning of Multipoint-to-Multipoint Communications in an MPLS ATM Label Switch Router

Author: Josep Mangues Bafalluy

Advisor: Jordi Domingo Pascual

DEPARTMENT OF COMPUTER ARCHITECTURE

UNIVERSITAT POLITÈCNICA DE CATALUNYA

A THESIS PRESENTED TO THE UNIVERSITAT POLITÈCNICA DE CATALUNYA

IN FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor en Enginyeria de Telecomunicació

MARCH 2003

Dr.	
President	

Dr.	
Secretari	

Dr.	
Vocal	

Dr.	
Vocal	

Dr.	
Vocal	

Data de la defensa pública	
Qualificació	

A la Montse,

A la meva família,

"Ever tried. Ever failed. No matter.
Try Again. Fail again. Fail better."

Samuel Beckett

Agraïments / Acknowledgements

En primer lloc, voldria agrair al meu director de tesi, Jordi Domingo, haver-me donat l'oportunitat de formar part del seu grup de recerca i haver contribuït d'aquesta manera a formar-me com a professional i com a persona. La seva paciència i afabilitat han estat de gran ajuda en la realització d'aquesta tesi.

També voldria donar les gràcies a la resta del grup i, en especial, aquells amb qui he tingut un tracte més estret, pel seu ajut sempre desinteressat, per tot el que m'han ensenyat i per les tertúlies dels dinars, moderades pel Carlos, que, amb el seu carisma, ha contribuït en gran mesura a la cohesió del grup.

Al departament, que m'ha donat l'oportunitat d'experimentar la docència universitària i la recerca i m'ha acollit en les diverses etapes de vinculació per les que he passat en tots aquests anys.

Als meus amics, per les bones estones i les muntanyes (físiques i metafòriques) que hem superat junts.

A la meva família, i en especial als meus pares i germans, pel seu suport incondicional, pel seu exemple, pels seus consells i per infinitat de coses que mai podré agrair suficientment. Estaré sempre en deute amb vosaltres.

I finalment, i de forma molt especial, a la Montse, per haver-se creuat en el meu camí i haver-lo fet més divertit, estimulant i intens. Hids.

Abstract

Multiprotocol label switching (MPLS) has appeared in the networking field to allow an efficient forwarding of packets through data networks. However, being a new technology, its main operational characteristics are based on those found in previous networking technologies. With regard to forwarding, the processing is based on label switching, which is also true for ATM networks, among others. Therefore, it seems logical to benefit from the expertise acquired in this field, especially as IETF has referred to ATM as one of the most likely link-level technologies to use in MPLS networks. Bearing this in mind, this thesis first focuses on the aspects that must be considered when offering multipoint-to-multipoint (or multicast) forwarding over ATM and then discusses their use in MPLS ATM Label Switch Routers (LSRs).

The main issue to solve for ATM multicast forwarding when using ATM adaptation layer 5 (AAL5) is the problem of cell-interleaving. In strategies using a multiplexing ID to solve it, there are two main options, namely source ID or packet ID. Our thesis argues for the convenience of using packet IDs because it allows the sharing of IDs among all the senders of the group, thus reducing ID consumption. We propose a new mechanism named Compound VC (CVC) that is characterized by making the ID variable in length to adapt to the group size and to reduce the overhead introduced by the multiplexing ID field. The implications of such design decision are discussed and evaluated.

As for implementation, store-and-forward (or VC merge) has been presented as the most likely mechanism to allow multicast forwarding in MPLS environments over an underlying ATM network. However, it presents extra buffering and delay, and it modifies traffic characteristics, which is not the case for CVC. The implementation issues of a CVC switch are also discussed. Though its implementation is slightly more complex than that of a legacy switch, it shows a similar level of complexity to store-and-forward. It is shown that CVC forwarding could be carried out in an MPLS ATM LSR by using state-of-the-art ATM hardware with slight modifications.

Resumen

La tecnología Multiprotocol Label Switching (MPLS) ha sido concebida para permitir un procesado eficiente de los paquetes en los nodos de las redes de datos. Sin embargo, y aunque se trata de una tecnología relativamente reciente, sus principales características operacionales se basan en otras que ya aparecían en tecnologías de red previas. Por lo que respecta al procesado de paquetes, éste se basa en la conmutación de etiquetas, principio ya utilizado en redes ATM, entre otras. Por tanto, parecería razonable intentar beneficiarse de la experiencia adquirida en este campo, más aún si tenemos en cuenta que IETF considera ATM como una de las tecnologías de nivel de enlace con mayor probabilidad de ser utilizada en redes MPLS. En este sentido, la presente tesis se centra, en primer lugar, en los aspectos a considerar cuando se ofrecen comunicaciones multipunto-a-multipunto (o multicast) en entornos ATM, y posteriormente, analiza su aplicación en MPLS ATM Label Switch Routers (LSRs).

El principal problema a resolver para ofrecer comunicaciones multicast ATM cuando se utiliza la capa de adaptación AAL5 es el entrelazado de celdas. En los mecanismos que utilizan un identificador de multiplexación para resolverlo hay dos opciones: identificar la fuente o identificar el paquete. Esta tesis argumenta en favor de la segunda opción puesto que permite la compartición de los identificadores entre todas las fuentes, y por tanto, se reduce el consumo de identificadores. En el presente trabajo se propone un nuevo mecanismo llamado Compound VC (CVC) que se caracteriza por utilizar identificadores de tamaño variable a fin de adaptarse al tamaño de cada grupo multicast y, al mismo tiempo, reducir el overhead debido a dicho identificador. Las implicaciones de tales decisiones de diseño son analizadas y evaluadas.

En cuanto a la implementación, store-and-forward (o VC merge) se ha perfilado como el mecanismo de implantación más probable en entornos MPLS-ATM. Sin embargo, dicho mecanismo introduce buffering y retardos adicionales y modifica las características de tráfico, cosa que no sucede con CVC. Los detalles de implementación de un conmutador CVC son también analizados. Aunque su implementación es ligeramente más compleja que la de un conmutador convencional, es aproximadamente igual a la complejidad introducida por store-and-forward. Se muestra que el procesado requerido

por CVC se podría realizar utilizando hardware ATM convencional con pequeñas modificaciones a fin de implantarlo en un MPLS ATM label switch router.

Table of Contents

List of Figures

List of Tables

Acronyms

AAL	ATM Adaptation Layer
ATM	Asynchronous Transfer Mode
CAC	Connection Admission Control
CBT	Core Based Trees
CDV	Cell Delay Variation
CRAM	Cell-Relabelling At Merge points
CT	Cut-Through
CTD	Cell Transfer Delay
CTU	Cell Transmission Unit
CVC	Compound Virtual Channel
DIDA	Dynamic IDentifier Assignment
DMVC	Dynamic Multiple VC-merge
FEC	Forwarding Equivalence Class
FMVC	Fixed Multiple VC-Merge
GFC	Generic Flow Control
ID	IDentifier
LAN	Local Area Network
LANE	LAN Emulation
LDP	Label Distribution Protocol
LPM	Longest Prefix Match
LSP	Label Switched Path
LSR	Label Switched Router
MARS	Multicast Address resolution Server

MCS	Multicast Server
mp2mp	multipoint-to-multipoint
mp2p	multipoint-to-point
MPLS	Multiprotocol Label Switching
MuxID	Multiplexing identifier
NBMA	Non-Broadcast Multiple Access
p2p	point-to-point
p2mp	point-to-multipoint
PDU	Packet Data Unit
PIM-DM	Protocol Independent Multicast – Dense-Mode
PIM-SM	Protocol Independent Multicast – Sparse-Mode
PNNI	Private Network-to-Network Interface
QoS	Quality of Service
SEAM	Simple and Efficient ATM Multicast
SF	Store-and-Forward
SMART	Shared Many-to-Many ATM reservations
SMVC	Selective Multiple VC-Merge
SPAM	Simple Protocol for ATM Multicasting
UPC	Usage Parameter Control
VC	Virtual Channel
VP	Virtual Path

Chapter 1

INTRODUCTION

This chapter introduces the context of this thesis and summarizes its motivations. It also gives a general introduction of what is the research problem tackled as well as the steps that were followed to solve it. In some sense, it may be understood as a brief explanation of what may be found in the thesis and how it is structured.

1.1 Introduction

MultiProtocol Label Switching (MPLS) is being developed to allow any layer 3 protocol to benefit from the fastest forwarding capabilities of switches when compared to traditional routers. Though the MPLS architecture of IETF allows any layer 2 technology to be used, ATM seems to be one of the most accepted due to the experience gained in ATM equipment design and the suitability of this technology to accomplish MPLS goals.

There have been some proposals to offer the multicasting service in MPLS, but most of them focus on signaling. This thesis focuses on the forwarding problems when offering MPLS multicasting using Label Switched Routers (LSR) whose underlying layer-2 technology is ATM.

First, we start by explaining why multicast communications are needed, what should be taken into account when offering multicast communications in general, and in particular, when using ATM. Notice that the word communication is used throughout this thesis as a generic term referring to both connection-oriented interaction and connectionless group interaction, which cannot be called connection as there is no establishment and tear-down.

The increase in the demand of multimedia applications (e.g. videoconferencing, interactive TV) and other group interactive applications (e.g. distributed simulation, cluster computing) has generated new problems that require rethinking some of the network architectures that have been used up to now. The need to provide communications inside a group is strongly associated with these kinds of applications.

In general, a multicast communication allows the exchange of information inside a group such that all the members of the group receive the information flow transmitted by the sender. Multicasting is often defined as a subset of broadcasting. In broadcasting, all the hosts in a given network receive the information though they did not ask for it, with the consequent unnecessary resource consumption in the network and in the hosts. The goal of multicasting is to selectively send the information to the hosts that asked for it.

The definition of multicasting based on that of broadcasting makes sense when broadcast media (or shared media) are used, because they have an inherent capacity to offer broadcasting. Shared media (e.g. ethernet) are especially suited to work with

connectionless network protocols, like IP. However, when considering a non-broadcast medium access (NBMA), like ATM, which is a connection-oriented technology, such definition does not make sense. In this environment, multicasting may be better defined as a superset of point-to-point communications, i.e. as multipoint-to-multipoint communications. Throughout this thesis, the term multicasting and multipoint communications are used interchangeably, unlike in some literature, where multicasting refers to the communication between one sender and multiple receivers, also known as point-to-multipoint communications.

IP solved the lack of multicasting support by designing an extension to the protocol named IP multicasting [Deering 89]. In ATM, some solutions to offer multicasting have been proposed. They are mainly focused on offering support to IP communications. They do that by introducing servers for information distribution and address resolution (e.g. IP multicasting over ATM [Armitage 97], or LAN Emulation [LNNI 99][LUNI 97]). In ATM, there have also been some proposals to offer the multicast service while fully exploiting the capabilities of ATM, e.g. quality of service (QoS) and transfer speed. This latter solutions try to offer the multicast service at the ATM layer, and thus, they avoid the complexity and inefficiency due to the interaction with the servers. We refer to the multicast service offered in this way as Native ATM Multicasting. However, the connection-oriented nature of ATM raises new problems.

Up to now, native ATM proposals satisfy some of the generic requirements for any multicast mechanisms and those particular to offer multicasting over ATM, but they still have drawbacks, e.g. scalability, suitability to group interactive traffic, and flexibility in group dimensioning.

Therefore, our research problem has been to find a native ATM multicasting mechanism that fulfills the requirements for multicasting in general, and multicasting over ATM, while it fully exploits ATM characteristics, provides efficiency in resource usage, and load-sharing between the entities involved in the multicast service.

Efficiency in the transport of multicast traffic over MPLS-ATM makes sense in an environment in which group interactive traffic is growing in importance. In ATM, unlike in IP, QoS was taken into account in the design process. As a consequence, this

technology presents characteristics that are particularly suited to this kind of traffic. But at the same time, transfer speed and scalability in the network are other points that would benefit these applications.

In conclusion, this thesis presents a new native ATM multicasting mechanism named Compound VC (CVC). It was designed with the requirements of multicast protocols in mind, and thus, it tries to solve the problems that appeared up to now. Our final goal has been to offer efficient multicasting at the MPLS-ATM level while allowing multimedia and other group interactive applications with stringent QoS requirements to benefit from the inherent capabilities of ATM in this area. The following sections introduce the steps followed towards this goal.

1.2 Problems of multicasting in ATM

The requirements of a generic multicast protocol [Braudes 93][Fahmy 97] might be used to classify the various problems to be solved when offering multicast communications in general, and in particular, making use of ATM. The issues to be addressed may be divided into a control part, which is usually software, and a forwarding part, which is usually hardware. The separation of the two functions allows both areas to improve in parallel without depending on one another.

As far as forwarding is concerned, ATM has proved to be a good switching technology in terms of scalability and switching speed, but some problems arise when attempting to offer multicast forwarding of cells through an ATM network.

Most current applications and technologies are designed to work over broadcast networks in order to fully exploit their potential. This is mainly due to the deployment of IP and its connectionless nature. On the other hand, ATM introduces new challenges to the multicasting problem due to its connection-oriented nature, as it is a Non-Broadcast Multiple Access (NBMA) technology, and also due to its QoS provisioning capabilities.

When establishing group communications, the routes to the members must be computed according to a requested QoS. Therefore, the signaling and routing protocol responsible for this (e.g. Private Network-to-Network Interface, or PNNI for short) is more complex than those found in IP multicast networks, which are based on best-effort

service. Furthermore, QoS must be enforced during connection establishment (through Connection Admission Control, or CAC) and during data forwarding (through Usage Parameter Control, or UPC). The problem is further complicated by the heterogeneous and dynamic nature of groups. These characteristics make it difficult to enforce QoS.

With respect to forwarding, the focus has been the transmission of IP multicast packets. In this case, the Adaptation Layer usually used is AAL5 [Grossman 99][Armitage 96]. The mechanisms and protocols designed up to now, such as VC Mesh and Multicast Server [Armitage 96], tend to imitate the operation of broadcast networks over connection-oriented networks. However, such solutions present important drawbacks, e.g. with regard to signaling overhead and scalability, and none of them offer native ATM multipoint-to-multipoint connections. For this kind of connections to be offered, the main issue to be addressed is the cell-interleaving problem. This problem appears at merge points of a shared tree through which multiple senders simultaneously transmit information packed into ATM Adaptation Layer 5 (AAL5) frames. These packets are usually longer than the payload of an ATM cell. Therefore, they must be fragmented into cells if they are to be transmitted through an ATM network. However, AAL5 does not have any multiplexing ID inside each cell indicating to which packet the cell belongs. This is not a problem for point-to-multipoint (or point-to-point) connections, but it is for multipoint-to-multipoint (or multipoint-to-point) connections. The problem only appears at merge points. At a merge point, two or more input virtual circuits (VC) belonging to the same multicast communication are forwarded through the same output VC. Therefore, the virtual circuit identifiers (VCI) of the input cells are mapped to the same output VCI. In this way, cells belonging to different packets may become interleaved at the output VC. Since AAL5 places no identifier (ID) inside each cell, the end-system will not be able to tell if a cell belongs to one packet or the other.

1.2.1 Current multicasting over ATM

The problem of offering multicasting over ATM while attempting to maintain the interoperability with connectionless environments has been studied by many organizations. The IETF has studied this issue in the Internetworking over NBMA (ION) and integrated services over specific link layers (ISSLL) working groups. At the ATM

Forum, the LAN emulation (LANE) and MultiProtocol over ATM (MPOA) working group and the multiway BOF also studied the problem, as did the study groups SG-11 and SG-13 at ITU-T.

There are two main models, namely the *overlay model* and the *peer model*. Key elements in the *overlay model* are separate addressing schemes, i.e. one entity has an IP and an ATM address and separate IP and ATM routing protocols and topologies. On the other hand, in the *peer model* all devices support a single address space, maintain a single topology and run a single routing protocol –all based on IP. This model has been realized through the development of MPLS mechanisms within the IETF. Earlier, the ATM Forum had worked on several initiatives to have homogeneous routing in heterogeneous environments.

In the first model, interoperability with current equipment is possible but the price paid is additional signaling overhead and delay in the data forwarding path. On the other hand, a peer approach scales better and is less complex, but the cell-interleaving problem must still be addressed. Furthermore, as ATM networks evolve and incorporate MultiProtocol Label Switching (MPLS) technology, it is likely that MPLS ATM Label Switch Routers (LSR) will employ a native ATM multicasting mechanism to address the cell-interleaving problem. All these points justify our research in native ATM multicasting mechanisms.

1.2.2 Native ATM multicasting

In the context of this thesis, native ATM multicasting refers to the mechanisms implemented at the switches to allow the correct ATM level forwarding of the information being exchanged by the members of a multicast group. That is, the cell-interleaving problem is solved without having to reassemble cells into AAL5 Packet Data Units (PDUs) inside the network.

Sometimes, VC merging is also used in the literature as a synonym for native ATM multicasting. However, this notation may be mistaken with the name of one particular mechanism, and thus, the expression Native ATM multicasting is preferred throughout this thesis.

Some of the mechanisms cited in the previous section solve the problem for legacy protocols by allowing them to interoperate in an ATM environment. Their technical approach is the imitation of a broadcast medium over a non-broadcast medium. As the focus is on interoperability, these mechanisms do not take full advantage of ATM characteristics, e.g. QoS provisioning and forwarding speed. If these characteristics are to be exploited, a different philosophy must be conceived.

Native ATM multicasting mechanisms aim to design a multicast strategy based on the ATM technology with no more restrictions, except the requirements all multicast protocols have. In this way, the capabilities of ATM could be fully exploited and efficiency in terms of resource consumption increases.

Native ATM multicasting techniques provide solutions for offering true multipoint-to-multipoint connections by solving the cell-interleaving problem at the ATM level, i.e. without any reassembly inside the network. True multipoint-to-multipoint refers to those group communications using a unique shared tree for all the members in the group.

Various solutions may be found in the literature to provide such communications at the ATM layer. A survey has been carried out and a taxonomy of native ATM mechanisms may be found in the following chapters. These mechanisms have been classified into four main groups, though a more rough approximation might allow to differentiate two main philosophies, namely those strategies that allow the multiplexing of cells belonging to different PDUs and those that avoid such multiplexing. Maybe the most representative one of the latter is what we refer to as store-and-forward (also known as VC merge), which has been proposed to be used in MPLS environments. However, it presents some drawbacks, mainly in terms of delay and buffering, that may preclude its usage in some representative cases, like multimedia applications.

The operation of the other proposals have also been analyzed and as a result, the unsolved problems have been identified. A new proposal that tries to solve these problems named Compound VC has been proposed. This mechanism is the main object of study of this thesis.

1.2.2.1 Compound VC (CVC) mechanism

The main goal of this thesis is to design a mechanism that could be applied in both local and wide area scenarios. It should also solve some of the problems of the mechanisms that appeared in the literature.

CVC allows the multiplexing of cells of different PDUs inside a single connection, and thus, traffic characteristics are not modified, unlike in other mechanisms. Furthermore, it provides flexibility in the sense that different group sizes and source characteristics may be accommodated while minimizing the required multiplexing overhead. This flexibility comes from the fact that the size of the multiplexing identifier may be negotiated at connection establishment.

However, in some cases, using one of the previous mechanisms may be appropriate. The flexibility of CVC allows to consider some of the mechanisms in the literature as particular cases. Furthermore, point-to-point connections can also be switched by using the same table as that used for multicast groups. Point-to-point connections are also a particular case of CVC communication.

In the evaluation carried out in this thesis, the reader may be able to confirm the advantages of PDU multiplexing identifiers of variable length. As explained above, the flexibility of CVC is based on this. Therefore, one aspect to consider for CVC is the dimensioning of such variable-length ID. The underlying idea is to find some relationship between the traffic parameters characterizing the sources with the probability of loosing a PDU due to not finding free multiplexing identifiers. Such rules would allow to dimension the PDU ID at CVC connection establishment given a PDU loss probability acceptable to the user. This evaluation has been carried out both analytically and through simulation.

Another important point this thesis tackles is the block-level design of a CVC switch. The idea is to study the increment in implementation complexity that comes as a consequence of introducing CVC in the forwarding process. As it is shown, one of the possible implementations may benefit from state-of-the-art ATM hardware with only slight modifications. Therefore, the CVC switch could help in leveraging current ATM equipment in the path towards MPLS. Furthermore, ATM seems to be accepted as one of

the most likely technologies to be used when implementing MPLS-LSRs. And thus, our work tries to make a contribution in this area.

The thesis is organized as follows. Chapter 2 provides a general introduction to multicasting and the problems that must be faced to provide such service. It also introduces the general background of the technologies related with the thesis by mainly focusing on the aspects of interest for our work. Particular emphasis is given to the issues to solve in MPLS-ATM environments.

Chapter 3 presents the background, which is more specific to this thesis. It describes the operation of native ATM multicasting proposals previous to our work. Their advantages and drawbacks are studied, and finally, a classification is provided.

The work carried out in Chapter 3 serves as initial discussion to Chapter 4, where the research problem of this thesis is stated and its validity and interest are discussed.

Our solution to the question raised in Chapter 4 is presented in Chapter 5. The design criteria followed for our native ATM multicasting proposal are presented. An example of its operation is also explained as well as some additional aspects that should be taken into account.

Chapter 6 is devoted to the evaluation of the diverse parameters that characterize the operation of Compound VC. This evaluation is carried out by mixing analytical and simulation results.

Chapter 7 explains the architecture of an ATM switch and an MPLS ATM-Label Switch Router that implements the Compound VC mechanism. An introduction to the building blocks of a legacy ATM switch is provided first, so as to discuss in following sections the modifications required with respect to it.

Finally, Chapter 8 summarizes the work carried out in this thesis, states its main conclusions, and presents some of the future lines of work.

Chapter 2

MULTICASTING

Multicasting has been a wide area of research since its conception due to the resource consumption benefits it offers. The support of multicasting has been mainly focused on IP. However, the provisioning of such capability in other technologies may also contribute in the final deployment of an all-multicast network. This chapter provides a brief description of what we understand as multicasting to define the environment of the thesis and the notation we adopted for the rest of the document. Next, the problems that appear when offering the multicast service are explained followed by some of the solutions that have been proposed. The last part of this chapter deals with MPLS, and particularly, with the interoperability or potential melding of both technologies for the provisioning of unicast as well as multicast communications. In this way, ATM benefits from the expertise in multicasting from the Internet community and IP benefits from the expertise acquired in ATM in switching, QoS, and traffic engineering.

2.1 Multicasting

This section presents a brief introduction on multicasting. Its definition and a review of the requirements of multicast protocols in general are presented.

2.1.1 Definition and Motivation of Multicasting

A generic definition of multicasting could be: 'Multicasting is the transmission of an information flow between two or more members of a group.' Without further restrictions, the scope of this definition is very wide. It would allow considering as multicasting any service offering such a transmission capability without regard of the layer at which it is done: Application, Network layer (IP), ATM layer, Physical (Ethernet).

For instance, one application willing to transmit to a group could trigger the establishment of a set of unicast connections, i.e. one-to-one, between the sender and all the receivers of the group. This communication scheme is represented in Figure 1.

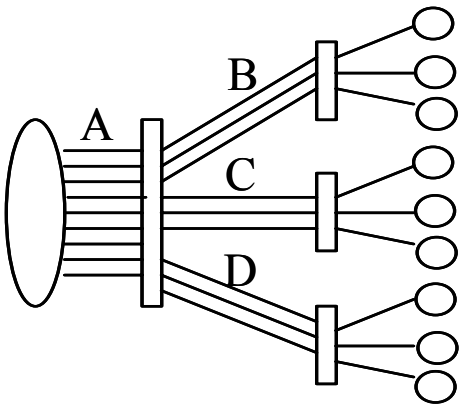

Figure 1. Multicasting at the application layer

The ellipse on the left of the figure represents the sender. The receivers are the nine circles on the right. The rectangles correspond to the routing/switching devices in the path between the communicating entities.

We name it as multicasting at the application layer because it is just the application and not the network that is conscious of the group relationship between the transmitter and the end-systems. Thus, the network just deals with unicast connections just as it did with one-to-one communications. Actually, this is just a particular case of a more general research

area named application-layer multicasting, application-level multicasting, overlay networks or some other similar names (see [Banerjee 02] for a review of such schemes).

In the example above, though the goal established in the definition is accomplished, remarkable problems arise when considering this scheme. We must notice that the same information flow is transmitted through each one of the nine connections between the sender and each receiver. That is, the same information is transmitted nine times by the sender through link A; it is routed/switched nine times by the first device, and three times by each of the three devices on the right. Eventually, just one flow arrives to each receiver. Thus, there is a remarkable waste of bandwidth in link A because one information flow instead of nine would be enough to make the information arrive to the receivers, and in links B, C, and D because one instead of three flows would be enough. Other drawbacks derive from the one just described. For instance, having nine connections implies nine times more processing and buffering requirements at end-systems and intermediate devices. Other more refined application layer multicast schemes are not as inefficient in terms of resource consumption as the one in the example. However, they do not solve this problem completely and some other concerns arise. On the other hand, network layer multicasting is much more efficient in this aspect, and this is the reason why we focus on these schemes.

In conclusion, application layer multicasting is inefficient in terms of resource consumption in the links, end-systems and routing/switching devices, and thus, it does not scale as network layer multicasting does. A more efficient way to offer group interaction is required. This is the motivation of network layer multicasting, which is usually referred to as simply multicasting.

If efficiency is measured in terms of network resources, it seems logical that the group consciousness should be managed at the layer where the routing/switching is carried out. This results in the communication scheme represented in Figure 2.

In this case, just one information flow passes through links that are common to more than one receiver and the whole group is treated by the network as a single communication and not as isolated unicast connections.

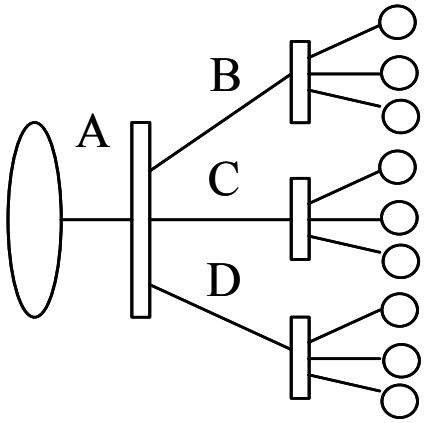

Figure 2. Network Layer Multicasting (or simply multicasting)

Bearing this in mind, the definition of what we consider as multicasting in this document must be reformulated with respect to the one given above. A more appropriate one is that found in [Armitage 97]: 'Multicasting is the delivery of a packet simultaneously to one or more destinations using a single, local transmit operation.' A single local transmit operation implies that the sender transmits the information just once and the network elements are in charge of replicating the information only when necessary.

Another more IP-oriented definition may be found in [Deering 89]: 'IP multicasting is the transmission of an IP datagram to a 'host group', a set of zero or more hosts identified by a single IP destination address.' As it is network-layer-oriented, it also shares the idea shown in Figure 2 and in the previous definition.

Therefore, network layer multicasting is more efficient in terms of resource consumption and more scalable than multicasting at the application layer in terms of the number of communications to be established by end-systems and the number of communications to be managed by network devices. Furthermore, scalability is further compromised when there is more than one sender in the group.

The growth in importance of group interactive traffic, like multimedia, is one example of the requirements imposed by new applications over the network. Apart from the provision of quality of service, maybe the most important challenge is to allow a user in a group to communicate with the rest of the group in an efficient way, as described above. In general, distributed multimedia applications consume a lot of resources and some of

them allow interaction between the members of a group, e.g. videoconferencing, cooperative work, and distributed interactive games. Therefore, scalability and efficiency are key aspects to consider when dealing with future applications.

Finally, we discuss some notational aspects around multicasting. Some words can be found in the literature that have similar or the same meaning as *multicasting*, e.g. *multipoint*. Usually, *multicasting* and *multipoint* are used interchangeably and that is how we will use them throughout this thesis. However, in a strict sense, multicasting is usually understood as a subset of broadcasting with the goal of allowing the transmission to a group of selected receivers and not to all hosts in a network. Thus, its usage would be restricted to networks based on broadcast media, like Ethernet. On the other hand, multipoint is understood as a superset of point to point communications. This notation is suited when working with connection-oriented networks, like ATM. A less common word also found in the literature to refer to *multipoint communications* is *multiway communications* [Jain 97].

Anyway, we will use them as synonyms, and we will mostly use multicasting to refer to multipoint-to-multipoint communications, not just point-to-multipoint, as it is found sometimes in the literature. And we understand multicasting as defined in [Armitage 97].

2.1.2 Multicast Protocol Requirements

We could use the requirements of generic multicast communications ([Braudes 93],[Fahmy 97], [Diot 97]) to classify the various problems to be solved when offering multicast communications in general, and in particular, when using ATM. The issues to be addressed may be divided into a control part, which is usually software, and a forwarding part, which is usually hardware. The separation of the two functions allows both areas to improve in parallel without depending on one another. In particular, ATM has proved to be a good switching technology in terms of scalability and switching speed, but some problems arise when attempting to offer multicast forwarding of cells through an ATM network.

The most important requirements for a multicast protocol, as stated in [Fahmy 97], [Braudes 93], and [Diot 97] are given below. Before transmitting information to a group, there must be some mechanisms that allow some hosts to be grouped under the same

group identifier without conflict. This is the function of multicast group address assignment mechanisms.

Concerning group set-up, apart from the unambiguous identification of a group, there must be other protocols allowing the allocation of an address for setting up a group. They should also allow the members of a group to know the address of the group they wish to join.

Another aspect to consider is the construction of the optimal multicast tree in the sense that it minimizes the number of network nodes where replication occurs so as to minimize resource consumption in the network.

Once the group has been initially established, membership management will allow hosts wishing to take part in a given group to join it, hosts wishing to end its membership to leave it, or to switch from one group to another.

Once the communication has been finished, some protocol must be defined to end the group communication, i.e. to carry out group tear-down.

The above points are related to group establishment and management and may be classified as connection establishment and maintenance, i.e. signaling, in a connection-oriented environment, or group establishment and maintenance in a connectionless environment. The following issues are more concerned with allowing the information transmitted among members of a group to reach the recipients.

Transport reliability is one of these points. Depending on the characteristics of the information transmitted to the group, reliability will take the form of error recovery at the receivers or retransmissions in case of losses. When retransmission is not possible due to time constraints, error recovery mechanisms will use redundant information to recover some losses. On the other hand, retransmission will be used when the focus is on the correctness of the information transmitted regardless of the time it takes to be transmitted.

Another aspect to consider is flow control. Its goal is to adapt to network load. It controls the information placed in the network per unit time and may therefore increase network efficiency by reducing the number of losses and consequent retransmissions.

Support for network heterogeneity is also required. Present networks are characterized by heterogeneous equipment at network nodes and end-systems, different network technologies, and different user requirements. This scenario makes adaptability a complex but necessary issue.

And last but not least, efficient packet forwarding strategies at network nodes must be developed to allow all the above requirements to be fulfilled. This thesis mainly focuses on this issue.

2.2 Multicasting in ATM Networks

Having discussed the generic characteristics of multicasting, this section is devoted to explaining the technical issues to consider when trying to offer the multicast service over an ATM network, the problems that appear, and some of the solutions that have been proposed up to now to solve them.

2.2.1 Problems of Multicasting in ATM

Most current applications and technologies are designed to work over broadcast networks in order to fully exploit their potential. This is mainly due to the development of Internet and related technologies, whose connectionless nature fits well in such environments. However, the inherent connection-oriented nature of ATM introduces new challenges to the multicasting problem in addition to those found in IP multicasting over broadcast networks. ATM is a Non-Broadcast Multiple Access (NBMA) technology. Consequently, multicast mechanisms cannot benefit from inherent broadcast facilities offered by a broadcast medium.

ATM introduces further challenges to the multicasting problem due to its inherent quality of service (QoS) provisioning and its connection-oriented nature. When establishing group communications, the routes to the members must be computed according to a requested QoS. Therefore, the signaling and routing protocol responsible for this (e.g. Private Network-to-Network Interface or PNNI) will be more complex than those found in IP multicast networks, which are based on best-effort service. Furthermore, QoS must be enforced during connection establishment (through Connection Admission Control, or CAC) and during data forwarding (through Usage

Parameter Control, or UPC). The problem is further complicated by the heterogeneous and dynamic nature of groups. These characteristics make it difficult to enforce QoS.

With respect to forwarding, the mechanisms and protocols designed up to now, such as VC Mesh and Multicast Server (see below), tend to imitate broadcast network characteristics over connection-oriented networks. However, such solutions present important drawbacks, e.g. with regard to signaling overhead and scalability.

As the focus of this thesis is on forwarding issues, it may be convenient to study how the information is carried through the ATM network. The adaptation layer that has received more attention for carrying data packets through ATM networks is ATM adaptation layer 5 (AAL5). The next section explains how it works. The problems associated with its use when offering multicast communications are explained in section 2.2.3.

2.2.2 ATM Adaptation Layer 5 (AAL5)

The information carried through an ATM network mainly comes from the layers above ATM in the form of packet data units (PDUs) which are usually longer than a cell. Therefore, a layer to adapt this information to the underlying transport is required. The ATM adaptation layer (AAL) [ITU-T I362] is in charge of making available to higher layers all the services that ATM can offer. It is divided into two sublayers:

> "The Convergence Sublayer (CS) is a service-dependent interface specification that can include multiplexing, error control, cell-loss detection, and timing recovery. The Segmentation and Reassembly (SAR) sublayer divides the variable length information from the CS into ATM cells for the ATM layer and reconstructs ATM cells into the original CS data units." [Chen 95]

The CS may be further subdivided into two sublayers, namely the common part CS (CPCS) and the service-specific CS (SSCS). The former provides unguaranteed connection-oriented transport of variable-length frames with error detection, whereas the latter provides functionality which is specific to each of the higher layers that may be carried with AAL5, and it may be null. As our focus is on forwarding of user data cells, SSCS is usually null, and thus, we focus on the common part.

Depending on the type of service required, the AAL chosen is different. When the AAL was first designed, four classes of service were considered, which were named A, B, C, and D. They were classified according to three main criteria, namely the timing requirements between source and destination, the bit rate, and the connection mode. One AAL was initially defined for each of these four types. AAL 1 and 2 were designed to support applications requiring guaranteed bit rates, while 3 and 4 were used for data traffic. However, AALs 3 and 4 were eventually merged into a single AAL named AAL3/4. Table 1 illustrates the relationship between the services and the AALs.

Table 1. Correspondence between classes of service and AALs.

<table>
<tr><th></th><th>Class A</th><th>Class B</th><th>Class C</th><th>Class D</th></tr>
<tr><td>Timing relation between source and destination</td><td colspan="2">Required</td><td colspan="2">Not required</td></tr>
<tr><td>Bit rate</td><td>Constant</td><td colspan="3">Variable</td></tr>
<tr><td>Connection mode</td><td colspan="3">Connection-oriented</td><td>Connectionless</td></tr>
<tr><td>AAL Protocol</td><td>Type 1</td><td>Type 2</td><td>Type 3/4, Type 5</td><td>Type 3/4, Type 5</td></tr>
</table>

As far as data traffic is concerned, AAL3/4 was first chosen, but it lost importance due to the remarkable overhead introduced. It added four more bytes per cell to the five bytes of the ATM cell header making it nine, which led to an efficiency of 83% (= (53-9) / 53 * 100). And this is without taking into account neither the CS Packet Data Unit (CS-PDU) header and trailer nor the padding. As a consequence, the new AAL5 [ITU-T I363.5] was designed to improve the efficiency and to allow a simpler processing. Despite the reduction of overhead fields, most functionality provided by AAL3/4 is still supported. For instance, the CRC-32 allows the detection of missordered or lost cells, as well as errors in bits; and the packet delineation is carried out through a bit in the header of the ATM cell [Peterson 00]. AAL5 has been widely adopted for carrying data traffic. Among the adopting organizations is the Internet Engineering Task Force (IETF), which recommends it to transport IP packets [Laubach 98] in ATM networks.

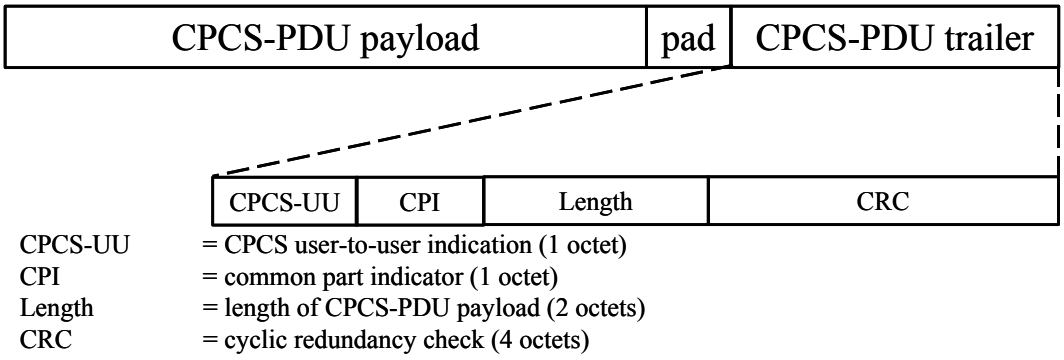

Figure 3. AAL5 CS-PDU format

The CS-PDU of AAL5 is presented in Figure 3. It supports up to 64Kbytes of data from higher layers. After the data portion there are two bytes in the trailer that are reserved for future use and two more bytes that carry the length of the data in the CS-PDU. The remaining four bytes carry the CRC-32. The SAR sublayer does not add more overhead to the cell but uses one bit in the Payload Type Indicator (PTI) of each ATM cell to determine the end of the PDU. This bit is the ATM-User-to-ATM-User indication (AUU), and when set to 1 indicates that the current cell is the last cell of the PDU. Otherwise, it is either the first cell or a cell in the middle of the PDU. The segmentation process is shown in Figure 4.

However, the multiplexing function provided by AAL3/4 by means of the Multiplexing Identifier (MID) field in the SAR-PDU is not available in AAL5. This field allowed PDUs coming from different applications to share a single Virtual Channel Connection (VCC). The MID field belongs to the AAL, which is not processed inside the network but at the end-node. Nevertheless, such field could help in the provisioning of multipoint-to-point or multipoint-to-multipoint connections if it was made available to switches, though it would imply a modification of the hardware and not all problems would be solved. Anyway, AAL5 has benefited from a wider acceptance, and thus, if the multicast service is to be provided, the potential cell-interleaving problem appearing at merge points of such connections must be solved. The cell-interleaving problem is further discussed in the following section.

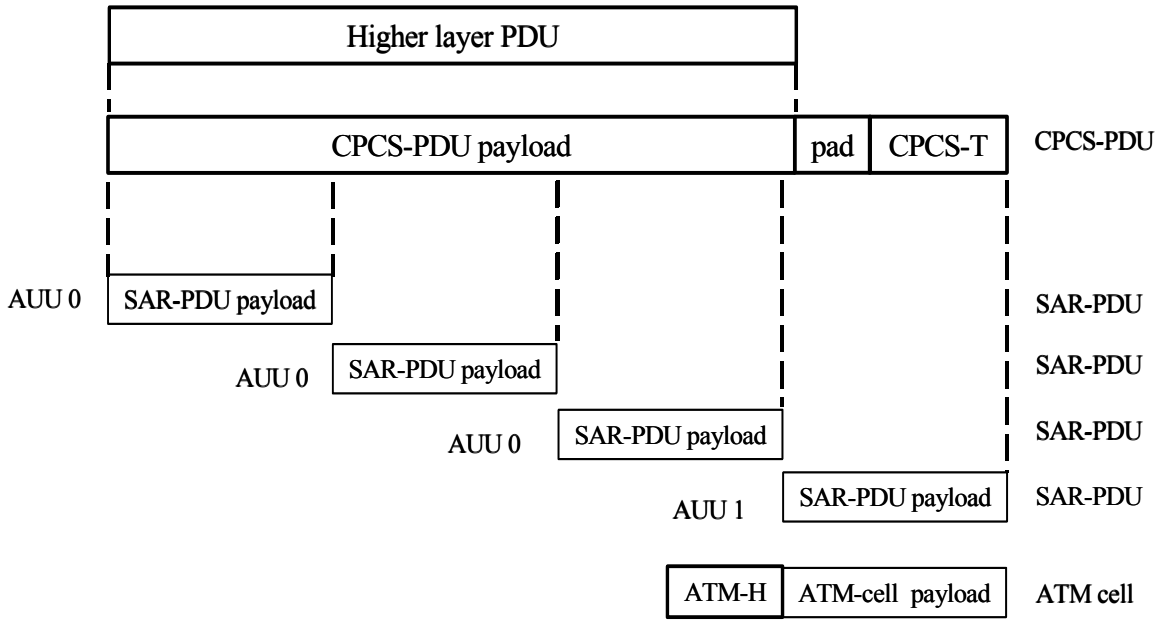

Figure 4. Encapsulation and segmentation for AAL5

2.2.3 The Cell-Interleaving Problem

The main issue to be addressed when dealing with the forwarding of AAL5 CS-PDUs through ATM networks is the cell-interleaving problem (see Figure 5). This problem appears at merge points of a shared tree through which multiple senders simultaneously transmit information packed into ATM Adaptation Layer 5 (AAL5) packets. These packets are usually longer than the payload of an ATM cell. Therefore they must be fragmented into cells if they are to be transmitted through an ATM network. However, AAL5 does not have any multiplexing ID inside each cell indicating to which packet the cell belongs. This is not a problem for point-to-multipoint (or point-to-point) connections, but it is for multipoint-to-multipoint (or multipoint-to-point) connections, as there are merge points in these latter connections. At a merge point, two or more input virtual channels (VC) belonging to the same multicast connection are forwarded through the same output VC. Therefore, the virtual channel identifiers (VCI) of the input cells are mapped to the same output VCI. In this way, cells belonging to different packets may become interleaved at the output VC. Since AAL5 places no identifier (ID) inside each cell, the end-system is not able to tell if a cell belongs to one packet or the other.

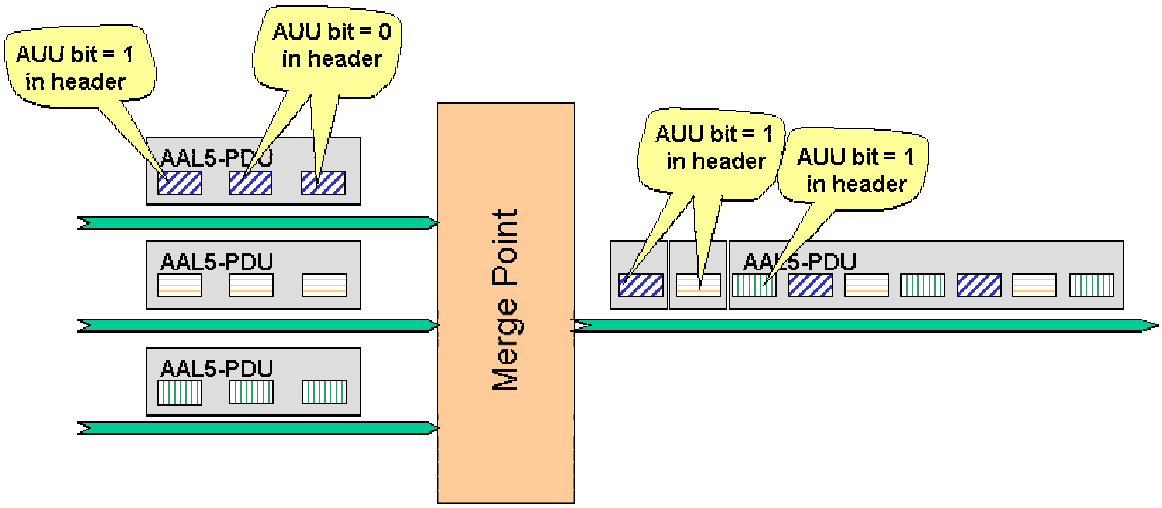

Figure 5. The cell-interleaving problem.

Figure 5 is an example of a merge point where three input VCs are forwarded through the same output VC. As explained in the previous section, in AAL5, one bit in the header of each cell is used to indicate whether this is the last cell of the packet (AUU=1) or not (AUU=0). If cells with the same output VCI became interleaved, the end-system considers that the first seven cells in the figure form a packet and the other two form a packet each.

The following sections explain the diverse solutions that have been proposed to solve the cell-interleaving problem. They seek to integrate IP and ATM in order to offer a multicast service. A different approach to the problem can be found in the next chapter, where proposals for native ATM multicasting are discussed.

2.2.4 Current techniques for multicasting over ATM: Integration of IP and ATM

This section provides a brief summary of current techniques being used to offer multicasting over ATM in the context of IP and ATM integration. This will allow to study their drawbacks and to justify the interest in exploring native forwarding approaches, which improve performance.

The problem of offering multicasting over ATM while attempting to maintain the interoperability with connectionless environments has been studied by many organizations. The IETF has studied this issue in the Internetworking over NBMA (ION) and integrated services over specific link layers (ISSLL) working groups. At the ATM

Forum, the LAN emulation (LANE) and MultiProtocol over ATM (MPOA) working group and the multiway BOF also studied the problem, as did the study groups SG-11 and SG-13 at ITU-T. This section provides a summary of this work.

There are two main models, namely the *overlay model* and the *peer model* [Alles 95]. Key elements about the *overlay model* are:

- separate addressing schemes, i.e. one entity has an IP and an ATM address, which are not algorithmically related, therefore
- IP to ATM address resolution is performed manually or dynamically via Address Resolution Protocols (ARP)
- separate IP and ATM routing protocols and topologies

A second model, the *peer model*, presumes that all devices support a single address space, maintain a single topology, and run a single routing protocol –all based on IP. This model has been realized through the development of Multiprotocol Label Switching (MPLS) mechanisms within the IETF. Earlier, the ATM Forum had worked on several initiatives to have homogeneous routing in heterogeneous environments, such as PNNI Augmented Routing (PAR) [PAR 99] and Integrated PNNI (I-PNNI) [IPPNI 96].

When offering the multicast service over ATM using the overlay model, there are two main options (IP multicasting over ATM –[Armitage 96], [Armitage 97]– and LAN Emulation –[LUNI 97], [LNNI 99]–). Interoperability with current equipment is possible without much effort, but the price paid is added overhead, and consequently, less efficiency. A brief description of both options follows. MPLS, which follows the peer model, is discussed in section 2.3.

2.2.4.1 IP multicasting over ATM

IP multicasting over ATM ([Armitage 96], [Armitage 97]) proposes a solution to support multicast communications in an ATM environment by using user-network interface (UNI) signaling version 3.1. Specifically, this makes use of point-to-multipoint connections to offer the multicast service. The first additional component is the Multicast Address Resolution Server (MARS). The MARS approach is built on the classical IP and ARP over ATM model, which provides an ARP and IP unicast service over ATM. MARS

offers a layer 3 multicast service with the signaling of UNI 3.1. The main function of MARS is to map IP multicast addresses to a list of ATM addresses.

There are two options for forwarding data to the members of a multicast group when using the MARS model, namely the *VC Mesh* strategy and the *Multicast Server (MCS)* strategy. The *VC Mesh* strategy establishes a point-to-multipoint tree from each sender to the rest of the members of a multicast group. In this case, a point-to-multipoint control VC from the MARS to the users of a group is required to notify membership changes. When such an event occurs, each point-to-multipoint connection from each sender to its receivers needs to be modified to add or remove the member. Therefore, the signaling load at end-systems is very high, and the time it takes for all the connections to be modified could be considerable. This strategy suffers from scalability problems and group stability latency is high. The main advantages of this approach are the distribution of the load across all switches of a network, and the optimization of the paths, minimizing end-to-end latencies as a result.

In the *Multicast Server (MCS)* strategy, multicast data distribution is centralized in a server. All the members of a group send data to this server. When a member wishes to transmit to the group, the packet is sent to the MCS by means of a unicast connection. All members may transmit at the same time; therefore, if AAL5 is used, the MCS must reassemble all cells belonging to the same incoming AAL5 packet data unit (AAL5 PDU) previous to its distribution through a point-to-multipoint connection. Cells belonging to the same packet must be transmitted together through the point-to-multipoint VC in order to avoid cell interleaving. The control connections between each member and the MARS and between the MARS and all the members are the same as in the previous option. The main advantages of this approach are: less resource consumption at end-systems, less signaling load between all entities involved in the multicast service, and less group stability latency. All these points are derived from the use of just one point-to-multipoint connection for data forwarding. However, it also presents some drawbacks: higher end-to-end latency due to buffering and non-optimal routes, and load concentration at the switches near the MCS.

2.2.4.2 LAN Emulation (LANE)

IP multicasting over ATM solves the problem for IP, but the rest of the network-level protocols (IPX, IPv6, NetBIOS, DECNet, AppleTalk, etc.) are not considered. LAN Emulation ([LUNI 97], [LNNI 99]) was designed to work in a multiprotocol environment, so that every application developed to work over any of the existing network-level protocols could benefit from the capabilities offered by ATM, and uses AAL5.

This technology is named after its main characteristic: it emulates a legacy LAN over a NBMA ATM network. In LANE, the ATM network is divided into different subnets called Emulated LANs (ELANs). Intra-ELAN unicast traffic is forwarded by means of point-to-point connections and broadcast traffic is served by means of the Broadcast and Unknown Server (BUS), which forwards multicast traffic to all the members belonging to the same ELAN as the sender. However, Inter-ELAN traffic is still forwarded through a router.

Focusing on multicast forwarding, the main components to consider are the Broadcast and Unknown Server (BUS) that already existed in LANE 1.0 and the Selective Multicast Server (SMS), which first appeared in the LANE Network-to-Network Interface (LNNI) specification of LANE 2.0. The forwarding procedure of both functional components is the same. The only difference is whether the message reaches all the members in an ELAN (as with BUS) or whether there are procedures to selectively choose a group of receivers in an ELAN (in SMSs).

BUSs and SMSs establish point-to-multipoint connections with the server itself as a root and the receivers as leaves. The sender wishing to send multicast information sends it through a point-to-point connection to the server. When this information arrives at the server, it is stored in reassembly buffers and reassembled. Once the destination is obtained, each packet is segmented into cells, which are transmitted back-to-back, thus avoiding cell interleaving.

In summary, the philosophy of multicast forwarding in LANE is the same as in MCS. As a consequence, the same advantages and drawbacks appear.

2.2.5 Native ATM Multicasting

From all that we have seen in the previous section we may conclude that IP Multicasting over ATM (overlay model) is problematic in terms of complexity, signaling overhead, or delay (see [Maher 97] and [Talpade 97] for an evaluation of some of these aspects). On the other hand, a peer approach scales better and is less complex, but the cell-interleaving problem must still be addressed. Native ATM multicasting mechanisms try to solve these problems by avoiding the processing at the AAL level, i.e. the processing of the cells carried out to offer the multicasting service will be done at the ATM level. The goal is to make the forwarding process more efficient.

Furthermore, as ATM networks evolve and incorporate MultiProtocol Label Switching (MPLS), it is likely that MPLS ATM Label Switch Routers (LSR) will employ some of the Native ATM multicasting mechanisms described in the next chapter to address the cell-interleaving problem. All these points justify research into native ATM multicasting mechanisms. As it is the field of research more closely related to this thesis, an entire chapter (Chapter 3) is devoted to describe the research that has been carried out in this field.

2.3 Multicasting in MPLS – ATM networks

Multiprotocol Label Switching (MPLS) is possibly the most successful example of application of the peer model and has recently deserved much attention. MPLS is designed to provide simplified forwarding, efficient explicit routing, traffic engineering, QoS routing, and not-trivial mappings of IP datagrams onto paths [Patzer 00].

Despite being a new technology, it is based on previous ones that set the first steps towards the final specification (e.g. Cisco's Tag Switching, Toshiba's Cell Switch Router). In turn, these technologies borrowed many concepts from previous technologies, especially ATM. In fact, most of the topics MPLS is focusing on have already been studied in ATM, and are some of its main characteristics. As a consequence, MPLS presents many similarities with ATM, and thus, it is interesting to study the paths from ATM to MPLS that will help in leveraging all the acquired knowledge in the ATM field in the recent past.

2.3.1 Multiprotocol Label Switching (MPLS)

MPLS stands for Multiprotocol Label Switching. Its main ideas are defined in IETF request for comments (RFC) 3031, which is entitled *Multiprotocol Label Switching Architecture* [Rosen 01] (this section is mostly based on this document). Its goal is to simplify the forwarding process of layer 3 packets through the network. The underlying issue is that layer 2 switches have simpler and faster forwarding processes than current routers. Therefore, the idea is to try to introduce such efficient processes for the forwarding of IP packets. Though any networking protocol could be used above MPLS, efforts mostly concentrate on the transport of IP, due to its supremacy over the rest.

In MPLS, packets are classified only at the ingress of the network into classes of traffic. This classes are called Forwarding Equivalence Classes (FEC), and refer to "a group of IP packets which are forwarded in the same manner (e.g. over the same path, with the same forwarding treatment)" [Rosen 01]. After the classification, ingress routers attach them a label and forward them to the next hop. The receiving node does not have to do longest prefix match (LPM) lookup operations, but table indexing based on a fixed-sized field (the label). To allow more flexible scenarios, label stacks were defined. By defining a hierarchy of labels, packets from an MPLS domain may be tunneled to a distant domain through another MPLS transit domain of higher hierarchy, and this transit domain is transparent to both edge domains.

Labels have local meaning and are negotiated by all the nodes involved in a given path through a Label Distribution Protocol (LDP). As a result of this negotiation process, a given FEC will have a Label Switched Path (LSP) established through the MPLS network that guarantees a specific treatment for that FEC.

Thus, the packet is not routed but switched according to the label that the upstream node attached to the packet. In this way, core nodes are freed from most of the burden due to LPM operations, and thus, it is claimed that this scheme scales better than usual IP forwarding. Routers that work this way are referred to as Label Switched Routers (LSRs).

The philosophy of MPLS is that of the peer model described in section 2.2.4. Therefore, all network nodes support a single address space, maintain a single topology and run a single routing protocol, based on IP. All these aspects, which could generically

be referred to as control, relate more to software. On the other hand, forwarding is more related to hardware. Therefore, control and forwarding are clearly separated in MPLS, which allows a parallel evolution of both areas.

Apart from a more efficient forwarding and scalability, MPLS is claimed to have some advantages over conventional network layer forwarding. For instance, as there is only one classification process at the ingress of the network, it could be complex and based on any field in the network layer header or the header of higher layers. And the complexity of the classification process is transparent to the rest of the LSRs, because they just switch labels. Another advantage is that MPLS supports traffic engineering in the sense that an LSP may be established that does not follow the path a dynamic routing protocol would choose. Another potential application is in the area of Virtual Private Networks (VPNs).

In light of what has been explained above, and according to the trends of the industry and the academia, MPLS seems to face a promising future. Therefore, it is worth it to work on defining the remaining aspects that were left for further definition in the MPLS architecture document. One of this aspects is multicasting.

2.3.2 Multicasting in MPLS

Apart from the generic advantages MPLS offers, multicast traffic presents additional characteristics that make it a strong candidate to benefit from MPLS [Dumortier 98]:

- Multicast flows often present a long duration and high bandwidth consumption, because they are usually related to multimedia communications. Therefore, a lot of layer 3 processing burden could be avoided if this traffic is switched.
- Detection of such flows is easy because they are usually set up using explicit signaling (e.g. PIM-SM), and thus, the mechanisms that trigger the establishment of a flow could benefit from this fact.
- Multicast flows usually use UDP as their transport protocol. Unlike TCP, UDP does not have congestion management mechanisms. This could mean that UDP traffic may have a negative effect on TCP flows, which reduce their transmission rate when the network becomes congested. If UDP traffic is switched, this effect is remarkably

reduced because layer-2 forwarding is more efficient than layer-3 routing. In this way, potential bottlenecks due to high volume UDP flows are reduced or avoided.

Multicasting is left for further study in the RFC that defines the MPLS Architecture. However, there have been many papers and IETF drafts covering multicasting. For instance, at the time of writing this thesis, [Ooms 02] is the most recent IETF draft that defines the *Framework for IP multicast in MPLS*. It mainly deals with the operation of current multicast protocols (e.g. PIM-DM, PIM-SM, CBT) in a general MPLS environment without much discussion of the particularities of specific layer 2 technologies, like ATM or frame relay. Thus, forwarding usually is either not covered or it is a secondary topic. And when dealing with multicasting, they are usually restricted to point-to-multipoint communications ([Acharya 97], [Dumortier 98], [Ooms 00], [Andrikopoulos 01]). Therefore, there are still some problems to solve if multipoint-to-multipoint communications are to be provided, especially when the underlying link layer is ATM.

2.3.3 Multicasting in MPLS – ATM

As explained above, ATM and MPLS have very similar characteristics as far as forwarding is concerned. As for forwarding, [Rosen 01] (p. 26) states that:

> "It will be noted that MPLS forwarding procedures are similar to those of legacy "label swapping" switches such as ATM switches. ATM switches use the input port and the incoming VPI/VCI value as the index into a "cross-connect" table, from which they obtain an output port and an outgoing VPI/VCI value. Therefore if one or more labels can be encoded directly into the fields which are accessed by these legacy switches, then the legacy switches can, with suitable software upgrades, be used as LSRs."

As a matter of fact, this was the philosophy of Toshiba's Cell Switch Router (CSR) [Katsube 97] and Ipsilon's IP switching ([Newman 97], [Newman 98]), which marked a milestone in the path towards the merging of IP and ATM. Shortly after, some more approaches following the same philosophy appeared, e.g. Cisco's Tag Switching, IBM's Aggregate Route-Based IP Switching (ARIS), though they have some specificities. A comprehensive survey of these technologies may be found in [Davie 98].

The important aspect of the previous quotation is that it paves the way for the application of some of the well-known concepts of ATM in MPLS. In fact, many of the main concepts of MPLS were already studied in ATM and are some of its distinctive characteristics (see Table 2). And some of the characteristics that are claimed an MPLS LSR should have to offer QoS make it more similar to an ATM switch (see Table 3). With the growth of group interactive traffic, the expertise acquired in ATM QoS may be another of the reasons to use ATM as the link layer [Dumortier 98].

Table 2. Parallelism between ATM and MPLS characteristics [Patzer 00]

	ATM	*MPLS*
Simplified Forwarding	VP / VC	Label
Routable Objects	Virtual Circuits	Label Switched Paths (LSPs)
Explicit Routing	Designated Transit List	Explicit Route Objects
Path Setup	PNNI Signaling	LDP-CR or Extended RSVP

The interest of the interworking and operation of both technologies has given as a result discussions like those in [AIC 01], [ITU-T Y1310], and [Alwayn 01], where potential scenarios for the evolution and integration of MPLS and ATM are described.

Focusing on forwarding, many ATM switch architectures have been recently designed and tested. If, as it seems, ATM switches and MPLS LSRs forward packets in the same way, there is a natural transition from one to the other. In this way, the ATM hardware can forward packets by switching the cells of which they are composed, as these cells carry the label in the VPI and VCI fields of their header. As for the software part, it should be changed, and the signaling protocols used in ATM should no longer be used. They are replaced by the control protocols of IP (e.g. PIM-SM).

Table 3. Non-ATM-LSRs should be similar to ATM switches to offer QoS [Patzer 00]

	ATM	*MPLS*
Queuing	Per-VC queuing	Per-LSP queuing
Traffic Scheduling	Weighted per-VC scheduling	Weighted per-LSP scheduling
QoS Routing	PNNI routing	Enhanced IGP (OSPF and IS-IS)

However, multicasting poses additional problems to ATM switching and to switching in general. Some multicast switch architectures have been proposed [Guo 98]. But

multicast is usually understood as the provisioning of point-to-multipoint communications. Nevertheless, the problem of the provisioning of multipoint-to-multipoint communications has not been solved, particularly if AAL5 is used. Native ATM Multicasting techniques aim to solve this issue for ATM switches, but according to what has been said, these solutions may also be applied to MPLS. This idea is reinforced if we take into account that ATM is one of the more common technologies used as link layer of MPLS networks ([Dumortier 98], [Andrikopoulos 01], [Armitage 00]).

2.3.3.1 Label encoding

Focusing on the details of MPLS - ATM operation when AAL5 is used, three label encodings have been defined in RFC 3031. SVC encoding uses the VPI/VCI field to encode the label. SVP encoding uses the VPI to carry the label on top of the stack and the VCI to carry the second one. And finally, SVP multipoint encoding uses the VPI to carry the top label, part of the VCI to carry the second one, and the rest of the VCI to identify the LSP ingress.

2.3.3.2 Label merging

Label merging is defined as the capacity to forward packets from the same FEC but coming with different incoming label values through a single outgoing label. This ability is fundamental if multipoint-to-multipoint communications must be provided, because if a single shared tree is used for all the group, merge points will appear. The problem is further complicated if, instead of full frames, the network is dealing with portions of frames carried in cells, as is the case for ATM. This has been the topic of discussion in section 2.2.3. RFC 3031 proposes two solutions to this problem.

VP merge, using SVP multipoint encoding builds multipoint-to-point VPs and distinguishes packets coming from different sources by assigning a different VCI inside the VP to each of them. In the second solution, named *VC merge*, buffers are required to buffer all the cells belonging to a single packet. Only after the last cell of the packet arrives, all the cells of the packet are moved in an atomic manner from the buffer where they were being reassembled to the output buffer for transmission. Though the first solution could be used with current equipment in some scenarios, it seems that VC merge

has deserved more attention due to its better scalability. There already exists some commercial equipment with the VC merge functionality [Cisco 01].

Label merging also has some implications in the negotiation of labels to use when an LSP is being established. They are discussed in detail in [Davie 01].

Though these proposals solve the multipoint-to-multipoint forwarding problem, they do not apply to all environments. This issue will be further discussed in the following chapters.

Chapter 3

STATE-OF-THE-ART OF NATIVE ATM MULTICASTING

Having dealt with the general background theory in the previous chapter, we now introduce the theory which is directly related with the topic of this thesis, that is, native ATM multicasting mechanisms. Unlike approaches that seek the integration of ATM and IP environments by imitating the behavior of a broadcast medium in ATM, native ATM multicasting mechanisms offer the multicast service at the ATM level. Therefore, a more efficient forwarding scheme is attained. A taxonomy of these kinds of approaches and an explanation of their characteristics is presented.

3.1 Introduction

In the context of this thesis, native ATM multicasting refers to the mechanisms implemented at the switches to allow the correct ATM level forwarding of the information being interchanged by the members of a multicast group. That is, the cell-interleaving problem is solved without having to reassemble cells into AAL5 PDUs inside the network, unlike in the Multicast Server (MCS) case [Armitage 97].

The mechanisms presented in section 2.2.4 solve the problem for legacy protocols by allowing them to interoperate in an ATM environment. Their technical approach is the imitation of a broadcast medium over a non-broadcast medium. As the focus is on interoperability, these mechanisms do not take full advantage of ATM characteristics, e.g. QoS provisioning and forwarding speed. If these characteristics are to be exploited, a different philosophy must be conceived.

Native ATM multicasting mechanisms aim to design a multicasting strategy based on the ATM technology with no more restrictions, except the requirements all multicast protocols have. In this way, the capabilities of ATM could be fully exploited and efficiency in terms of resource consumption would increase.

Native ATM multicasting techniques provide solutions for offering true multipoint-to-multipoint connections by solving the cell-interleaving problem at the ATM level, i.e. without any reassembly inside the network. True multipoint-to-multipoint refers to those group connections using a unique shared tree for all the members in the group. Though we use the expression native ATM multicasting to refer to these techniques, sometimes VC-merging (or VC-Merge) is used in the literature for the same purpose ([Baldi 98], [Chow 99]) and sometimes it is used to refer to a particular mechanism ([Schmid 98], [Widjaja 99]), which corresponds to Store-and-Forward (SF) in our classification. Based on the preliminary classification in [Baldi 98], Table 4 presents a more comprehensive classification [Mangues 00c] with some notational changes, new mechanisms, and new groups of mechanisms added with respect to that from [Baldi 98].

Techniques belonging to the first type solve the cell-interleaving problem by avoiding cells from different packets to be interleaved. They are generically referred to as

strategies that *Avoid cell-interleaving*. *Buffering* techniques reassemble all the cells of each PDU in separate buffers and forward them without mixing cells belonging to different buffers (or PDUs) ([Grossglauser 97], [Rosen 01], [Stolyar 99]). Shared Many-to-many ATM ReservaTions (SMART) uses a *token passing* scheme to allow just one sender to put data in the multicast tree at any instant [Gauthier 97]. In the second type (*VP switching*), the virtual path identifier (VPI) identifies the connection and the virtual circuit identifier (VCI) of the ATM cell header is used as the multiplexing ID (identifying the PDU [Calvignac 97] or the source: [Venkateswaran 97], [Baldi 98]). The third type of strategies *Allow multiplexing inside the same VC*. This is done either by *adding overhead* in the transmitted data ([Komandur 97], [Komandur 98], [Baldi 98]), by using the *Generic Flow Control (GFC)* field in the header of the ATM cell [Turner 97], or by negotiating, at connection establishment, the sequence with which cells are going to be transmitted to the downstream node [Chow 99]. Finally, in *Multiple VC techniques*, two or more VCIs are used as *PDU IDs* [Venkateswaran 98] or *group IDs* [Venkateswaran 98] for the same multicast connection.

Table 4. Classification of native ATM multicasting mechanisms

Avoid cell-interleaving		VP switching		Allow multiplexing inside a VC			Multiple VC switching	
Buffering	Token control	Source ID	Packet ID	Added overhead	GFC	Signaling	Group ID	Packet ID
Cut-through (SEAM) CT-NC CT-T Store-and-Forward	SMART	VP switching VP-VC switching VP switched CLIMAX	DIDA	SPAM CRAM AAL5+ based CLIMAX	Subchannel (WUGS)	VC-merge scheduler	FMVC	DMVC SMVC

A more detailed description of these mechanisms may be found in the following sections.

3.2 Avoiding Cell-Interleaving

The main advantage of these approaches is their scalability in terms of number of groups. There can be as many multicast connections as there are VCs available, because only one VC is used for all the traffic of the group.

3.2.1 Buffering

The first type of techniques solves the cell-interleaving problem by avoiding cells from different packets to be interleaved. This is referred to as cut-through (CT) forwarding. Simple and Efficient ATM Multicast (SEAM) [Grossglauser 97] is one example of these techniques. It buffers the cells of a packet until no other cells are being forwarded to the same output VC. This buffering, when carried out at each switch in the path, presents the additional effect of increasing burstiness, latency and cell-delay variation (CDV); as a result, traffic characteristics may be violated. The term cut-through forwarding means that the forwarding of a PDU starts when the first cell of the PDU arrives if the outgoing VC is idle and continues until the last cell of the PDU arrives. Therefore, a slow source could block the rest of the sources for significant time intervals. The store-and-forward (SF) proposal follows the same idea, but in this case, the first cell of a packet is not forwarded until all the cells of that packet have been buffered (see Figure 6). Once the last cell arrives, all the cells of that packet are buffered together in the output buffer where they wait to be transmitted consecutively. This latter approach seems to be the most likely implementation when ATM is used in MultiProtocol Label Switching (MPLS) environments [Rosen 01]. In this thesis, we will jointly refer to SEAM and store-and-forward as *buffering* techniques.

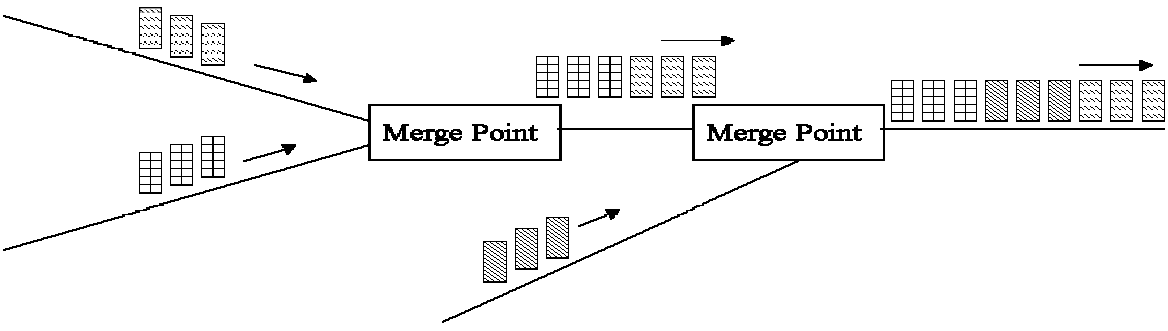

Figure 6. The store-and-forward strategy avoids cell-interleaving by buffering all the cells of a packet prior to its transmission.

An extension of buffering techniques for providing some kind of quality of service (QoS) classes is briefly discussed in [Widjaja 98] and [Widjaja 99]. The proposal of the authors is to use different output buffers for traffic requiring different QoS. Each output buffer is assigned a different VC and a cell level scheduling mechanism is responsible for interleaving cells of different classes in order to minimize traffic distortion.

A stability study of CT strategies may be found in [Stolyar 99]. The authors argue that CT techniques using a round-robin scheduling discipline to assign the buffers are not stable when the packet rates of all input VCs are not the same. In this context, instability means that the content of the buffers grows without any bound. Some variations are proposed to solve this problem. For instance, cut-through-on-no-contention (CT-NC) does not forward a partially arrived packet if there is a complete one in another queue. As a result, CT-NC combines the delay reduction of traditional cut-through and the stability of store-and-forward.

Cut-through-with-thresholds (CT-T) allows its queues to dynamically change from CT to SF. For any given queue, there are three working options. First, the queue performs CT forwarding while the number of complete packets in the other queues of the same connection is below threshold H1. Second, if the number of packets is above H1, it performs SF forwarding for this queue. Third, it returns to CT operation when the number of packets in the other queues is below another threshold H2.

The results obtained in their simulations show a significant improvement in delay for CT-NC with respect to CT. They also show, in general, a slight improvement with respect to SF, though in some cases this improvement could be greater.

The main advantage of buffering strategies is their simplicity. Though these techniques may allow easier implementation when compared to the other types, their main drawback is the buffering requirements at the switch. This buffering, when carried out at each switch in the path, presents the additional effect of increasing burstiness, latency, and CDV with the result that traffic contract may be violated. As a consequence, their main application would be data transmissions, and not real-time transmissions. The solution proposed in [Widjaja 99] to offer some kind of QoS to store-and-forward does not entirely solve this issue. Problems derived from buffering will remain for traffic

belonging to the same class. Therefore, if the class granularity is not very small, i.e. if there are not many different buffers, one could expect that all the group interactive traffic will pass through the same buffer and its traffic parameters will therefore be distorted. Furthermore, if a large number of classes are defined to allow cell interleaving between classes, VC consumption will increase. Therefore, the scalability advantage claimed by buffering strategies over other strategies is diminished.

3.2.2 Token Control

In Shared Many-to-many ATM ReservaTions (SMART) [Gauthier 97] cell-interleaving is avoided by means of a token passing protocol that allows only one sender to put information in the shared tree at any given time. In this case, the shared tree is accessed as if it were a shared medium. This mechanism allows the enforcement and accomplishment of the traffic contract, because enforcing the contract of the group at any given time corresponds to the enforcement of the sending end-system.

The main advantage of SMART is that the management of the QoS offered to the whole group is reduced to solving the problem for the sender that is transmitting in a given instant.

But SMART is a complex protocol because all switches must interchange Resource Management cells in order to allow the token to move from sender to sender, which imposes a considerable overhead. The mechanism is further complicated if more than one tree has to be managed simultaneously in order to allow some senders to send to the group at the same time. Therefore, complexity in group management leads to scalability problems.

3.3 VP Switching

In VP switching techniques, the VPI identifies the group and the VCI is used as the multiplexing ID. A further subdivision differentiates between the VCI that identifies just the packet and the VCI identifying the source. Dynamic IDentifier Assignment (DIDA) [Calvignac 97] follows the former scheme, while [Venkateswaran 97] deals with the latter and proposes a modification, which is named VP-VC switching. The modification seeks to combine the advantages of what the authors call VP switching and VC switching. VP

switching in [Venkateswaran 97] imposes a globally unique VCI identifying the sender. In addition, what the authors call VC switching corresponds to a strategy with the VCI value being changed at each switch. Therefore, additional mechanisms are required to identify the sender. The VCI mapping is static and once established, it lasts until there are no more cells coming with a given input VCI.

VP switched CLIMAX [Baldi 98] is another VP switching mechanism using source ID in the same way as in [Venkateswaran 97]. CLIMAX stands for CelL-Interleaved Merged ATM conneXions.

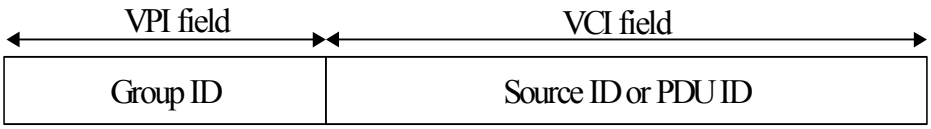

Figure 7. VP switching.

Though these strategies could be implemented with small or no modifications to current switches, their main drawback is lack of scalability in terms of the number of groups that can be established. The VPI field has just 8 bits at the user-network interface (UNI) and 12 at the network-network interface (NNI). Lack of group size flexibility is another disadvantage. For instance, in DIDA all the VCIs of a VP are assigned to a group even if the group is small, because the VP identifies the connection. Thus, having 2^{16} identifiers in a group is the smallest granularity. Moreover, the advantages of using packet IDs as multiplexing IDs is not fully exploited in DIDA, because the identifiers are of a fixed and large size. As a consequence, the efficiency in ID consumption due to using packet IDs is lost due to a large number of IDs being unused for a given connection.

The utilization of the VPI field of the cell header as group ID may also represent a problem if carriers attempt to use it for other purposes.

Another drawback for VP switching source ID strategies is that the implementation of Early Packet Discard (EPD) is difficult because switching is carried out using the VPI, without keeping any state information for the VCIs inside the VPI.

3.4 Allowing Cell Multiplexing inside a VC

The underlying philosophy of these strategies is to use just one VC for each multicast connection. There are three main ways to do this: by adding extra overhead to the cell to carry the multiplexing ID, by using the GFC field to carry the multiplexing ID, or by negotiating the multiplexing order of cells through signaling. The following subsections describe each of these strategies in more detail.

3.4.1 Added Overhead

In this group, we classify those techniques that propose a modification of AAL5, and particularly its segmentation and reassembly PDU (SAR-PDU), by adding an extra field that carries multiplexing information for each cell. Again, this field could be used to identify the packet or the sender. In the latter case, a global ID assignment is needed. Simple Protocol for ATM Multicasting (SPAM) [Komandur 97] is an example of these techniques that uses per source IDs (see Figure 8). The same approach is followed by AAL5+ based CLIMAX [Baldi 98].

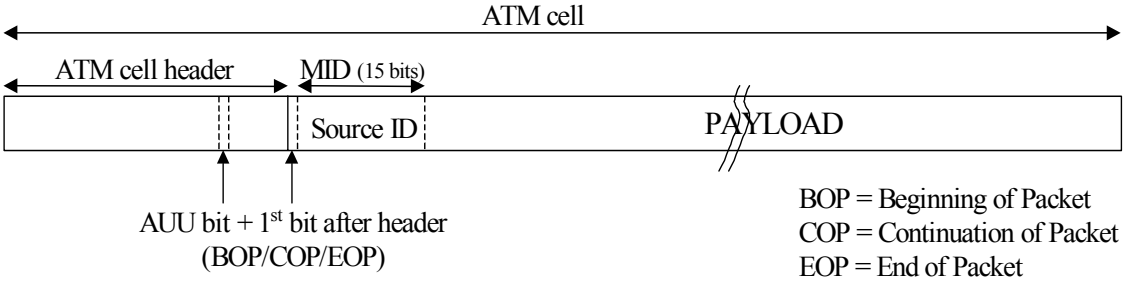

Figure 8. Simple Protocol for ATM Multicasting (SPAM): Cell format.

We have grouped together several techniques as Added Overhead to make a more generic definition. Therefore, we include Cell Re-labeling At Merge Points (CRAM) [Komandur 98] as one particular case. In CRAM, the multiplexing IDs for each cell are related with the source that transmitted the cell and these IDs are locally remapped at each merge point. However, unlike in SPAM, the multiplexing IDs are not carried inside each cell. In this case, they are carried in Resource Management (RM) cells, which precede a block of interleaved cells (see Figure 9). At the first merge point, an RM cell (and its corresponding block) is built by assigning source IDs to all the cells from the same input

VC that are to be transmitted through the same output VC. When a block arrives at a merge point, the multiplexing IDs are extracted from the cell, they are remapped, and a new block is built by buffering the cells from all the merging blocks.

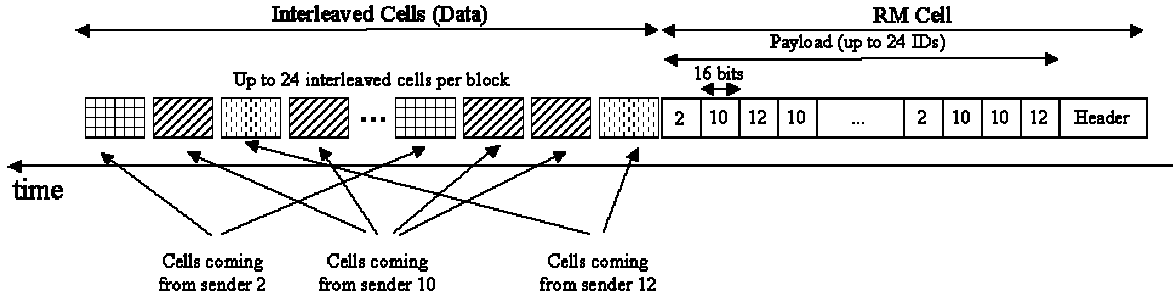

Figure 9. Cell Re-labeling At Merge Points (CRAM): Block format.

The problem of flexibility of group size also arises in SPAM and CRAM, where multiplexing identifiers have fixed length. As in these cases the VCI (not the VPI) identifies the connection the flexibility problem as such is not as important as in the other cases.

In addition, these techniques include an overhead that reduces the bandwidth available to user information on a given link.

Furthermore, in SPAM, the switch needs to perform AAL processing by looking at the multiplexing ID in the SAR-PDU, a task that, in principle, corresponds to the end-system. In the case of CRAM, RM cells are processed in the switch. This processing should be added to the table look-up operations. Moreover, the processes of analyzing and creating a block consisting of the RM cell followed by the cells indexed by it, needs some buffering and it would affect latency, CDV, and burstiness, though this effect is not as harmful as in *buffering* mechanisms.

Another drawback is that the multiplexing ID in the payload is not protected by the HEC error detection/correction capability, which only covers the cell header. Therefore, an error in a multiplexing ID may affect cells from sources using different IDs.

3.4.2 Generic Flow Control (GFC) used as Multiplexing ID

Another mechanism that cannot be classified within one of the three types above is proposed in [Turner 97]. The multiplexing identifier, called the subchannel identifier, is

carried in the GFC field of the ATM header. That gives the possibility of fifteen simultaneous packets in a switch (one ID is left to indicate an idle subchannel). It identifies a burst or packet formed of data cells delimited by RM cells. The subchannel ID is dynamically changed at each switch for each packet. With this strategy, there is no extra overhead due to multiplexing information, but there is some due to the insertion of RM cells, though it could be avoided if no reliability concerns are imposed and the end-of-packet indication of AAL5 is used to differentiate the packets. This mechanism is implemented in the Washington University GigaSwitch (WUGS). Turner also proposes using more than one VC if fifteen IDs are not enough to cope with all the traffic, and establishing a block of VCs (one per subchannel ID) that are shared by all senders solves interoperability with non-subchannel switches.

One of the main problems of this mechanism is the utilization of the GFC field, which means that it will not be available at the UNI for user access, e.g. in a passive optical bus, or for other purposes. Moreover, it could limit the potential widespread use in NNI interfaces, as there is no GFC in these interfaces.

3.4.3 Signaling

In this case, the cell-interleaving problem is solved by negotiating at connection establishment the order in which cells from the different queues at an upstream node will be transmitted downstream. [Chow 99] presents a scheduler capable of offering native ATM multicasting (or VC-merge as the authors call it) with a 2-level hierarchical structure based on the output module of a per-VC queuing ATM switch. These switches determine the order in which each VC is served by means of a scheduler. This scheduler is preserved in their proposal. The second level is introduced through the concept of the Virtual Queue. A Virtual Queue consists of a sequencer and a set of subqueues for different incoming VCs having the same outgoing (merged) VC. Recall that a VC corresponds to a multicast connection in strategies that allow cell-interleaving inside a VC.

Therefore, if both the sequencer at the upstream node and the de-sequencer at the downstream node know this order, cells belonging to different packets may be correctly separated. The authors argue that their proposal allows the provisioning of per-flow QoS

to both VC-merge and non-VC-merge connections. Furthermore, their work shows that it is fair and that it presents a bounded delay for the worst case.

However, the operation of this scheduler may not be correct under low traffic conditions. The snooping mechanism used to solve this problem may solve the sporadic lack of a cell but not the continuous lack of cells from different Virtual Queues. Moreover, the snooping mechanism requires a modification in the utilization of the VPI, as the authors propose the use of the least significant bit of the VPI to mark snooper cells. Finally, the connection establishment procedure becomes more complicated due to the negotiation of the sequencing of cells.

As a general remark, the main advantage of strategies that allow multiplexing inside a VC is their scalability in terms of number of groups, because just one VC per group is used. However, this scalability comes at the price of extra overhead, limited group size, or extra signaling complexity.

3.5 Multiple VC Switching

A new proposal that did not appear in the original classification [Baldi 98], called Multiple VC switching, has been added. Multiple VC switching shares some characteristics with *VP switching* and *Allow cell multiplexing*.

Multiple VC switching is similar to VP switching in that the multicast connection is not a single VC, but multiple VCIs are assigned to the same connection. Multiple VC switching techniques are similar to SPAM and the like because cell multiplexing is allowed in the connection, but in this case not in a single VC but in a compound connection composed of several VCs.

The purpose of these strategies is to reduce the buffer requirements at the cost of increased utilization of VPI/VCI space. This should not cause scalability problems as there are usually far fewer connections than VCIs in any given link [Venkateswaran 98].

There is a further subdivision of these kinds of mechanisms. Some mechanisms use the VCIs as packet IDs and others use them as identifiers for a group of senders.

3.5.1 Group ID

In Fixed Multiple VC-merge (FMVC) [Venkateswaran 98] each sender is assigned a fixed connection ID at a merge point. Alternatively, some senders may share this ID, which will consequently identify a group of senders. Therefore, this mechanism allows the interleaving of cells belonging to different groups of senders.

FMVC may use the multiplexing IDs more efficiently than source ID mechanisms. However, efficiency may be significantly reduced in cases where there is a very active group and many inactive groups. In this case, it is possible that while the inactive groups of senders are not using their identifiers, the active group would run out of IDs, and thus, cells belonging to packets that did not get an ID need to be buffered. In conclusion, there is a lack of flexibility in the assignment of IDs plus an additional complexity in the implementation of the buffering.

3.5.2 Packet ID

In this case, the VCIs assigned to the multicast connection are used as packet IDs. In Dynamic Multiple VC-merge (DMVC) [Venkateswaran 98], each switch maintains a set of unassigned IDs at each outgoing link pertaining to a given connection. When the first cell of a packet arrives, one ID is assigned and is maintained for all the cells of the packet. When there are no free IDs, cells are stored until one ID becomes free.

Selective Multiple VC-merge (SMVC) [Venkateswaran 98] is an enhancement to FMVC and DMVC for dealing with store-and-forward (SF) native ATM multicasting. When using two IDs, one is used for SF forwarding and the other one for cut-through (CT) forwarding in the following way. If an entire packet has been received, it is forwarded through the SF ID. Cells of partially arrived packets use the CT ID.

These mechanisms do not represent an improvement in terms of throughput with respect to SF native ATM multicasting, but they have great impact on groups using CT. Furthermore, DMVC obtains a buffer reduction of 80% with two connection IDs, when compared to cut-through. Finally, some simulation results also showed that SMVC requires 50% less buffer then SF if two IDs are used.

Despite of having these advantages, some drawbacks also appear. For instance, the implementation of DMVC shows the additional complexity of the management of the dynamic assignment of IDs at output ports to each PDU. In addition to this, there are extra buffering requirements, because cells of PDUs waiting an ID to be freed when all of them are in use. In this case, cells are buffered in some entity similar to the reassembly buffers that appeared in SF (or VC-merge) switches, with the problems commented above with respect to group interactive traffic. As a result, the additional complexity of DMVC switches increases with respect to conventional ATM switches due to the management of IDs and of the reassembly buffers. [Venkateswaran 98] says nothing about this complexity and the modifications that must be done in state-of-the-art switches to support DMVC. Furthermore, it is not clear what number of IDs will be assigned to each connection, and what are the parameters that determine the choice.

As for SMVC, though 2 IDs may be enough for small groups with low volume of traffic, they may not suffice for multicast groups using group interactive applications. Thus, this approach is not flexible in terms of group size (number of members) and volume of traffic that could be handled. Besides, there is also the additional complexity of the reassembly buffer management, which makes it look like an SF switch, as in the DMVC case. In fact, the authors claim that SF is a particular case of DMVC or SMVC in which just one VCI is assigned to the connection.

3.6 Summary of Native ATM multicasting mechanisms

The advantages and drawbacks of all the strategies for offering Native ATM Multicasting are compared in Table 5.

ATM was designed to provide high switching and transmission capacity, while offering the requested quality of service (QoS) for each connection. Multicast connections will usually carry group interactive information with real-time requirements. Consequently, the provisioning of a mechanism that fully exploits the characteristics of ATM while offering an efficient multicast service would be an important step towards widespread multicast deployment. In particular, the emerging MultiProtocol Label Switching (MPLS) implementations of MPLS ATM Label Switched Routers (LSR) might benefit from the multicast forwarding approaches presented in this chapter.

Table 5. Advantages and drawbacks of native ATM multicast forwarding.

Type	Subtype	Advantages	Drawbacks
Avoid cell-interleaving	Buffering	Scalability (number of groups) Easy implementation No additional overhead for mux IDs	Buffering requirements Increased burstiness, latency, and CDV
	Token control	Scalability (number of groups) Easy group QoS provisioning	Difficult connection management Signaling overhead
VP switching	Source ID	Easy implementation (no hardware modification) No additional overhead for mux IDs	No scalability (number of groups) No flexible mux ID size Carriers may use VPI Difficult EPD implementation
	Packet ID	Easy implementation (slight hardware modification) No additional overhead for mux IDs Efficiency in mux ID usage due to packet ID	No scalability (number of groups) No flexible mux ID size Carriers may use VPI
Allow multiplexing inside a VC	Added overhead	Scalability (number of groups) Reduced buffer requirements	High overhead No ID size flexibility AAL or RM processing in the switch Mux IDs not protected by HEC (CRAM) buffering -> increased latency, CDV, and burstiness (SPAM) modification of standard AAL5
	GFC	Scalability (number of groups)	GFC only available at the UNI
	Signaling	Scalability (number of groups) Per-flow QoS (bounded delay) Fairness	Bad operation in low traffic conditions Connection establishment complexity Uses one bit of VPI
Multiple VC switching	Group ID	Better ID usage than strategies with fixed size mux ID No additional overhead for mux IDs Group size flexibility	Increased VPI/VCI space utilization Inefficient operation under some conditions
	Packet ID	Better ID usage than strategies with fixed size mux ID No additional overhead for mux IDs Efficiency in mux ID usage due to packet ID Traffic characteristic unchanged (QoS) Mux ID size flexibility	Increased VPI/VCI space utilization Complex implementation

If we take a look at current ATM networks, AAL5 is the most commonly used adaptation layer, even for group interactive and computer supported cooperative work (CSCW) applications. If AAL5 is to be used in multicast connections, the most important issue to solve for multicast forwarding mechanisms is the cell-interleaving problem. Native ATM multicasting includes those mechanisms that solve such problems at the ATM layer, without the need for higher layer processing at the switches or at servers. Four main groups of solutions are explained, namely: avoiding cell interleaving, VP switching, allow multiplexing inside a VC, and multiple VC switching.

In the first solution, there are two alternatives: 1) Buffering strategies are simple to implement but modify traffic characteristics, and thus, their application to group interactive communications could be restricted, and 2) Token control mechanisms have simple traffic contract management but complex connection management.

The main drawback of VP switching strategies is lack of scalability in terms of number of groups, especially when the identifiers are assigned to the sources.

The third type presents three alternatives: 1) Added overhead shows scalability in terms of number of groups, but at the price of extra overhead, 2) GFC uses the GFC field, only available at the UNI, which could limit its potential deployment in a wide area, and 3) Signaling determines the order in which cells from VC-merged connections are going to be transmitted.

Finally, multiple VC switching presents flexibility in group size, no extra overhead, and respects traffic characteristics at the price of a more complex implementation of the switch.

Chapter 4

THE NATIVE ATM MULTICASTING PROBLEM

Chapter 4 is devoted to state the problem this thesis tackles in a detailed manner. The assumptions made and the main points it focuses on are also introduced. After that, the justification of the validity of the question is presented. The discussion in previous chapters on the behavior of the existing mechanisms and their drawbacks is taken up again for that purpose jointly with other considerations that justify that the question is still unanswered. Finally, there is a discussion on why is it a worthwhile question.

4.1 Introduction

The growth in importance of group interactive traffic demands an efficient support of multicasting in terms of both forwarding and control. This efficiency may come from the side of ATM and MPLS if some of the problems for offering multicasting in this environments are solved. The following sections introduce the problems to solve and state the question this thesis will study. In that sense, it separates the work previously carried out by other authors from our proposal.

4.2 Statement of the problem

In light of what has been said and the discussion that has been scattered through the previous chapters, the main problem solved in this thesis may be stated as follows:

To design and evaluate the forwarding part of a native ATM multicasting mechanism which 1) solves the cell-interleaving problem when AAL5 is used, 2) copes with most of the problems that appear in previous proposals, and 3) has application in MPLS ATM LSRs, with special emphasis on the incidence in group interactive communications.

The following paragraphs explain in more detail the whole meaning of the statement and how each of the terms is understood throughout the thesis.

The design process took into account all the previous knowledge on the field and the conclusions of the comparative analysis of all the mechanisms. As a result, the benefits and drawbacks were highlighted and the argumentations the authors of each of the mechanisms gave for the design choices they took were taken into account. This led us to define the requirements and design criteria for our proposal.

The evaluation is done analytically and through simulation, in which we build a merging scenario and study the various parameters characterizing the mechanism proposed in this thesis. In this sense, the approach was similar to other proposals that appeared in the literature. The results obtained serve to study the dependence of the behavior of the mechanism on the traffic parameters of the sources that inject traffic into the merge point.

As previously explained, the topic of multicasting is very broad and involves different aspects that must be considered to offer a complete solution, e.g. group set-up, maintenance, and tear-down, address assignment, or flow control. This thesis tackles one of these topics: forwarding. This term is understood throughout the thesis as the process by which a packet is received in an input port, the fields of its header are processed, and with the help of a table, a decision on what output port (or ports) the packet should be sent through is eventually made. Notice that the table we referred to in the last sentence, which tells the path a packet should take according to the values of the fields in the header, is filled by another process called routing, which is not included in forwarding. This is also the way it is usually understood in the networking community, e.g.[Peterson 00] (p. 264). According to the OSI model, forwarding is the main function of the network layer. However, with the increased importance of switches, and particularly, since the appearance of ATM, some of these functions were moved to the more efficient hardware-based layer-2. That is the reason why forwarding is sometimes referred to as switching. On the other hand, routing has more to do with control, and is implemented in software. Therefore, what is discussed in this thesis is more related to hardware.

Our aim is to design a native ATM multicasting mechanism, expression that was introduced in the previous chapters. Recall that this expression is used to generically refer to the mechanisms that offer the multicast forwarding capability at the ATM layer. They are called native mechanisms, because all the required processing is carried out at layer-2, i.e. there is no reassembly of cells into AAL5-PDUs. And according to our conception of multicasting, they are multicast because they offer multipoint-to-multipoint capabilities. Multipoint-to-point communications are also included because they are a particular case of the more generic multipoint-to-multipoint ones. This is in contrast to what is often found in the literature, in which multicast and point-to-multipoint communications are used as synonyms.

A consequence of considering multicasting in this manner is that merge points may appear. Consequently, cells coming from different input branches to the merge point may be interleaved. If the more common adaptation layer for data, i.e. AAL5, is used, the receiver is not able to correctly reassemble the flow of interleaved cells, because there is no multiplexing identifier in each cell that tells to what PDU it belongs. This is known as

the cell interleaving problem (section 2.2.3), and it is the main problem native ATM mechanisms try to solve, and so does ours.

The second requirement for our proposal is that it must try to solve the problems that appeared after the analysis of the characteristics of each of them, which were discussed in Chapter 3. The following section argues on this topic.

The relationship between MPLS and ATM has also been discussed in section 2.3. As one of the emerging technologies in the networking field, MPLS has received much attention lately. With respect to forwarding, it is widely accepted that ATM and MPLS behave in the same way. Therefore it is likely to think that the work carried out in the forwarding field in ATM could be transferred to MPLS. Therefore, one of our goals is to contribute to leverage the knowledge acquired in ATM in various areas (e.g. QoS, switching) in the path towards MPLS.

All these goals should be accomplished bearing in mind the ever increasing importance of multimedia and other group interactive traffic. This kind of traffic poses requirements to the network that are more strict than other services which are more insensitive to delay-related parameters. As most group interactive applications are inherently multicast, this is an essential aspect to consider.

4.3 Validity of the question

The discussion on the mechanisms presented in the previous chapter served us to observe that though they solve the cell-interleaving problem, there are still some other problems to solve, particularly if these mechanisms are used with group interactive applications.

Mechanisms that provided interoperability between IP and ATM had either much signaling (VC mesh approach) or extra delay due to reassembly processing and buffering (MCS approach), which might make both options impractical for group interactive applications ([Maher 97], [Talpade 97]). Their main problem came from the fact that multicasting, i.e. multipoint-to-multipoint communications, was not provided at the ATM layer. Other mechanisms that offer multicasting capabilities in an MPLS-ATM environment (sections 2.3.2 and 2.3.3) provide them just using point-to-multipoint

connections, so the same problems as the VC mesh approach arise. Besides, the use of point-to-multipoint trees prevents bi-directional shared trees, because this latter trees have merge points, with which these mechanisms are not able to deal.

Therefore, native ATM multicasting mechanisms are preferred because, as they offer multicasting at layer-2, they are more efficient and applicable to group interactive environments. A brief overview of the main advantages and drawbacks of each mechanism was presented in Table 5. Here we will discuss the main points that justify the need for future research in this area.

The most representative buffering strategy is store-and-forward (or VC-merge). ATM Forum has accepted VC-Merge to support multipoint-to-point connections in future versions of PNNI specifications [Venkateswaran 98]. IETF also points out that VC-merge is one of the possible strategies to use [Rosen 01]. However, it presents two main problems, namely extra buffering and modification of the delay characteristics of traffic. As for the former point, though there is some work that states that for the scenario the authors tested there is a small increase in buffer [Widjaja 99], there are other papers that state the contrary [Zhou 99]. In general, it is usually accepted in the literature that there may be a substantial increase in the buffering requirements when using store-and-forward (SF). With respect to the latter, this buffering translates into a modification in the delay characteristics, e.g. cell transfer delay (CTD) and cell delay variation (CDV) at the cell level, and of end-to-end packet delay and inter-packet delay variation at the packet level. This effect was studied in [Boustead 98] where the delay characteristics for a real traffic trace were measured. The authors concluded that the delay for hardware VC merge was 65% higher across the range of network load until 85%. Standard deviation also was 95% higher in the hardware VC merge case, and this increase stayed approximately constant up to about 85% utilization.

Therefore, there is still the need for a mechanism that does not increase the buffering in the nodes, and as a consequence, does not vary the delay characteristics of the traffic so that the contract could be respected. Hence, we agree with [Venkateswaran 98] in the sense that “ATM networks, which support statistical multiplexing, derives some of its

advantages from the interleaving of cells. Therefore, it may not be desirable to prevent cell interleaving."

As for token passing schemes, a simplified connection management is required, because there is much overhead due to control messages used to manage the sending turns. Furthermore, schemes that just allow one sender at a time may be useful for some applications but not for those that require all partners in the group to send and receive at the same time with full interactivity, e.g. videoconferencing.

VP switching mechanisms show scalability problems because they use the VPI to identify the group. Thus, a small number of simultaneous groups that pass through a given link at the same time may be established, even more if we take into account that some VPIs are required for unicast traffic. Besides, the VPI may be used by carriers for other purposes, thus preventing its use for group identification. Furthermore, in some cases, the field they use for the identification of the source or packet may be overdimensioned, because they use the 16 bits of the VCI. There may be some applications requiring such huge number of senders or simultaneous packets, but it is not the usual case. Therefore, there is still the need for a mechanism that provides some flexibility in group size without the waste of bits in the header VP switching strategies impose.

The drawbacks of strategies that allow multiplexing inside a VC depend on the way they control interleaving. The main claim against the first subtype (added overhead) is that they add extra overhead to the already high ATM overhead. Furthermore, identifiers are not carried in the header, and as a result, the mechanisms implemented in ATM to detect header errors do not protect them and may affect the traffic from other sources in case of errors. With respect to the second subtype, the GFC is only available at the UNI, and it is just four bits long, though the proponents argue that more than one connection could be used if needed (albeit adding extra connection management complexity). As for the last subtype (signaling), apart from the extra complexity in the signaling process due to negotiation of the cell sequence, there is the snooping mechanism. Some problems appear owing to this mechanism. First, as it uses one bit in the VPI, the same problems as in VP switching appear with respect to VPI utilization. And second, the mechanism does

not work properly if there is an interval during which some source sends slow traffic. In conclusion, it would be good to find a mechanism which tries to solve most of these problems. Particularly, no extra overhead should be added and it should be available in any interface (UNI or NNI).

As for multiple VC mechanisms, they show an additional complexity in the implementation. It comes from two main points. First, the usage of more than one VCI for the same multicast connection, which requires a negotiation of the number required. But nothing is said in [Venkateswaran 98] about neither how to do it nor the modification in current switches to support these mechanisms. And second, switches will still have to manage reassembly buffers in the same way SF-capable switches do. Therefore, depending on the number of VCIs chosen for the multicast group, the same buffering and delay requirements as in buffering strategies may appear.

Apart from the comments specific to each of the mechanisms, a general remark on packet ID vs. source ID strategies is in order. There are some of the previous strategies which use either one option or the other. However, there has not been an evaluation of the advantages of one over the other.

All the problems discussed above also apply to MPLS if one of such mechanisms is used at the ATM level to provide multicast capabilities. But there is still one to solve concerning the label encoding. As seen in section 2.3.3.1, the IETF proposes three different encoding possibilities. Therefore, there is a need to study these possibilities and the proposal of a new one if it is required to fulfill the requirements imposed over our proposal.

In light of what has been said, we think there is still room for a new proposal that provides multicast services that takes into account the particular characteristics of group interactive traffic and adapts to an MPLS-ATM environment.

4.4 Is it a worthwhile question?

In our opinion, multicasting has always been a worthwhile question. As discussed in section 2.1.1, the provisioning of efficient mechanisms for multicasting is interesting in the sense that it allows a better usage of network resources. But this efficiency should not

come at the price of punishing multimedia and other group interactive traffic, because it is already important and forecasts claim that it will grow in importance in the near future hand in hand with broadband technologies deployment [Stordahl 02]. Furthermore, most multimedia applications are inherently multicast (e.g. videoconferencing, on-line gaming, CSCW tools). Moreover, the availability of high bandwidth in the links implies the transition from a pull-oriented philosophy (e.g. web retrieval) where the user searches the net and asks for some content towards push-oriented and multi-party philosophies focused on the general public. The distribution of contents in these latter schemes should take advantage of multicasting for scalability reasons. And, as the contents is multimedia traffic with explicit group memberships and predictable flows, it fits better with a connection-oriented environment, where, in addition, QoS contracts may be more easily enforced [Dumortier 98].

Focusing on specific technologies, MPLS multicasting has not yet been solved. The RFC describing the MPLS architecture [Rosen 01] left it for further study. Though the definition of a 'framework for IP multicasting in MPLS' is under discussion in the IETF, there is not yet an RFC [Ooms 02]. This draft is mainly generic in the sense that it does not consider any particular layer-2 technology, except for a small section that refers to ATM and Frame Relay. In fact, the use of ATM as the layer-2 technology poses further unsolved problems. For instance, the merging problem stated in the draft, which is due to the fact that these technologies do not support multipoint-to-multipoint or multipoint-to-point connections, thus preventing the merging of LSPs [Ooms 02]. Therefore, native ATM multicasting mechanisms may help to solve this problem, whose solution would also lead to a more efficient use of the label space. Besides, ATM is one of the most likely used layer-2 technologies in MPLS networks mainly due to two factors, namely the similarities in the operation between both technologies and the expertise acquired in ATM in various fields that are also of interest in the MPLS community (e.g. QoS and switching). In conclusion, finding a mechanism that provides an efficient multicasting scheme (i.e. multipoint-to-multipoint, thus solving the merging problem) able to support the stringent requirement group interactive applications pose, may be a key component in the leveraging of the acquired ATM knowledge that will ease its application in MPLS environments.

Therefore, we think that the provisioning of a multicast forwarding mechanism might help in transferring some ATM concepts to MPLS to solve problems that were already solved for ATM and that are under study in MPLS.

Chapter 5

THE COMPOUND VC MECHANISM

Having seen in previous chapters that there still are some unanswered questions in the field of native ATM multicasting research, this chapter tries to provide a contribution to the aforementioned points that remain unsolved. It introduces a new native ATM multicasting mechanism called Compound VC switching (CVC). The design criteria are also introduced, as well as an explanation of the way they determine the behavior of CVC. These criteria determine the operational characteristics of the mechanism and the way cells are handled. To this effect, a birds-eye view of the forwarding of multicast traffic with CVC is also given, though it is covered in depth in Chapter 7. The new mechanism is also classified into one of the types introduced in previous chapters and compared to previous mechanisms by means of an example that illustrates the main differences among them.

5.1 Introduction

The previous chapter served us to justify the need for research in the native ATM multicasting area with application to MPLS. Therefore, it introduced the motivation to propose a new mechanism whose operation is explained in this chapter. Its main goal is to solve some of the problems other mechanisms have, with special emphasis on group interactive traffic.

A series of previous assumptions are made throughout this chapter and also the thesis. As it is mainly devoted to forwarding, it is assumed that there exist signaling and routing capabilities that allow the creation of shared bi-directional multipoint-to-multipoint trees with the appropriate QoS requirements, though e.g. the latest version of PNNI does not yet support such connections [PNNI 02].

This chapter is organized as follows. First, the design criteria for the new mechanism are presented. Next, we explain the operation of the mechanism and the way traffic forwarding is done in CVC scenarios. Following that, a qualitative comparison with the rest of native ATM multicasting mechanisms is presented. And finally, some additional comments on CVC operation are given.

5.2 Design criteria

Section 4.3 pointed out the main drawbacks found in the previous proposals. This section states the main characteristics imposed over the new mechanism.

5.2.1 Allow the multiplexing of cells

The classification presented in Table 4 is based on the way each strategy employs to solve the cell-interleaving problem. Except for the mechanisms labeled as *Avoid Cell-Interleaving*, which use buffers or a token, all the rest use some kind of multiplexing ID (muxID) to deal with this problem. Multiplexing IDs allow cells belonging to different PDUs to be interleaved, and thus, the traffic characteristics of all the sources in the group are respected. However, in *buffering* strategies, the buffering requirements and the resulting increase in CDV and burstiness limits its application to group interactive communications. The connection management complexity of *token*-passing schemes like

SMART may also limit its application to group interactive communications. Therefore, time-constrained traffic may be more suitably served by allowing multiplexing of cells belonging to different PDUs. The price paid by current muxID strategies is that they usually add some extra overhead to carry the ID, which adds to the intrinsic ATM overhead. But this problem may be solved if the right position of the muxID is chosen.

Therefore, the new mechanism should allow the multiplexing of cells. There is some work in the literature that justifies this decision. For instance, [Boustead 98] carried out some measurements with real traffic traces to compare the delay behavior of the store-and-forward (VC-merge) mechanism, which does not allow the multiplexing of cells, with the VP switching (VP-merge), which does allow it. Their results showed that there was a substantial increase of both average delay and delay variation, which are critical for group interactive applications. This increased delay comes as a consequence of the increment of the buffer required to provide the reassembly functions required in VC-merge. Figure 10 presents the delay distribution the authors obtained. The effect on the average delay may be observed in the drift to the right of the curve labeled as *Hardware merge* with respect to that labeled as *No hardware merge*. The variation in the delay also increases, as may be observed from the fact that the *Hardware merge* curve is wider than the other. Other studies in the same direction but with simulated traffic were carried out in [Venkateswaran 98] with similar results.

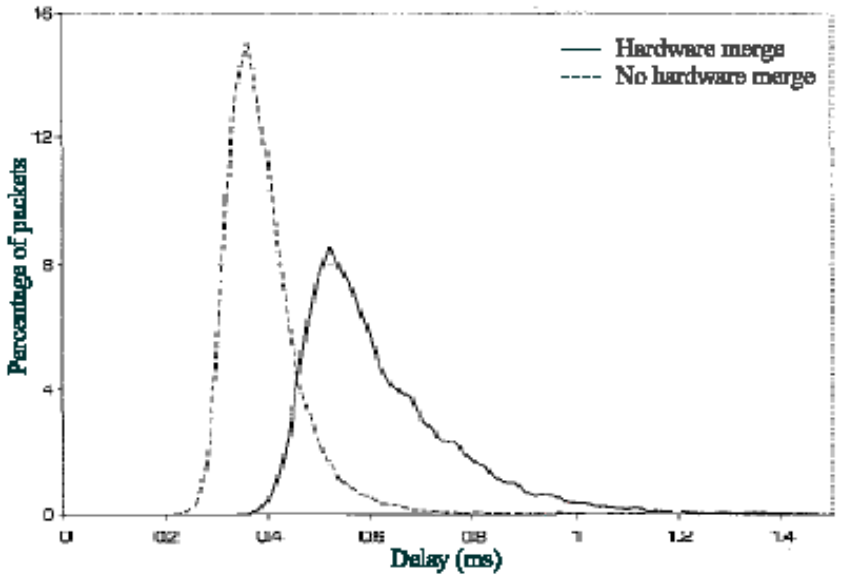

Figure 10. Delay distribution for the real traffic trace used in [Boustead 98] at 60% load

These results argue for allowing the multiplexing of cells. But there is still a more philosophical argument that comes from the conception of ATM. As a matter of fact, ATM derives some of its advantages from the statistical multiplexing of cells. But AAL5-PDU store-and-forward mechanisms do not follow this scheme.

In conclusion, for the environment which constitutes our target, and also for philosophical reasons, it seems logical to provide the means to keep the multiplexing of cells for multicast connections. But in this case, if AAL5 is used, the cell-interleaving problem must be solved by inserting a multiplexing identifier in each cell.

5.2.2 Prefer Packet ID to Source ID

For those approaches in Table 4 that allow the multiplexing of cells, a more rough classification may be established depending on whether source or packet multiplexing IDs are used. Both options allow the receiving station to differentiate the cells that belong to one PDU from those that belong to another one. But there are some implications behind this decision that we discuss below.

In Source ID mechanisms, the ID is related to the source that transmitted the packet. Therefore, there is a binding between the source and the ID at each switch. This binding must be unique at each switch to avoid ID collision at merge points. The ID collision problem may be solved either by globally assigning IDs for the group or by locally remapping the IDs at each switch [Venkateswaran 97]. The management required in the former option may limit its scalability because a central authority must determine the uniqueness of identifiers. On the other hand, local remapping maintains a list of free IDs at each switch, where a local mapping of IDs is carried out, and then, the identifier received by the end-host is not the same the sender wrote in the cell.

Source ID mechanisms usually overdimension the size of the ID to solve the worst case in which there could be a lot of senders. Usually, these kinds of mechanisms use 15 or 16 bits for the identifier, and thus, traffic from up to 2^{15} and 2^{16} senders may be multiplexed. However, not all groups will have such a huge number of senders, and most overhead will be unused, e.g. in the local area. In addition to this, some mechanisms use additional fields apart from those in the header of the cell to carry the identifier. As a consequence, the overhead is increased.

On the other hand, PDU ID strategies assign an ID to each packet, and this assignment is independent of the source this packet came from. A new incoming PDU to the switch is assigned an ID from a pool of free IDs. Thus, packets coming from all the sources in the group share the identifiers and the identification of the source is a responsibility of higher layers. In this way, ID consumption is smaller than with Source ID.

In conclusion, the sharing of IDs of packet ID strategies may help in reducing the overhead required in other mechanisms if accompanied by flexible ID size negotiation, as explained below. Furthermore, no scalability problems arise due to the global ID assignment process from a central authority.

5.2.3 Negotiation of the size of the identifier

In some of the mechanisms, either source ID or packet ID, a fixed-sized identifier is used. Often, in source ID mechanisms, it is large to accommodate a big number of sources in the worst case, as explained above. As for packet ID mechanisms, fixed-sized fields are also usually used, and consequently, they do not fully benefit from the ID sharing of packet ID strategies. For instance, DIDA uses a 16-bit field, which is overdimensioned, even more than in the Source ID case, because all the senders share these IDs. WUGS subchannel mechanism, on the other hand, uses small fixed-sized IDs (the 4 bits of the GFC field in the ATM cell header), which may be insufficient for bigger groups. In this case, more than one GFC-connection should be used and the group management is then increased.

Furthermore, the size of the groups and the traffic characteristics of the traffic sent by their members may substantially vary from one group to the other. For example, if we are using packet IDs, the sharing of identifiers is different if there are a lot of senders transmitting at low rates and when there are few senders transmitting at high rates.

Using fixed-size overdimensioned identifiers might be an adequate solution in environments with a lot of spare bandwidth that may be used by some additional overhead, but that would make the communication more inefficient. On the other hand, the processing of fixed-length fields is simpler. But in an ATM environment, which is already characterized by its *cell-tax,* overhead should not be inefficiently used. Therefore, we think that the solution to this problem consists of providing a way to negotiate the size

of the identifier so that it is able to merge all the traffic generated by the senders in the group with its particular traffic characteristics, thus avoiding the overdimensioning of identifiers with the consequent loss of bits in overhead that is not used.

5.2.4 No additional overhead

If multiplexing of cells is allowed, and thus, there are merging points where cell-interleaving may occur, each cell should carry a multiplexing identifier somewhere. One possible solution would be to modify AAL5 to add an additional field in SAR-PDU (e.g. SPAM), in the same way AAL3/4 does. But one of the main concerns about AAL3/4 was the excessive overhead it introduced, which adds to the already high overhead of ATM. Consequently, if overhead is one of our concerns this is not a valid solution.

The ideal situation would be that in which no extra fields are added to the SAR-PDU to carry the identifier. Therefore, the only solution is to carry it inside the header of the cell in a similar way to VP switching strategies. But unlike in these latter strategies, if this is combined with a flexible ID size negotiation (see section 5.2.3), the space of identifiers will be fully used and it will not be partly wasted due to overdimensioning.

5.2.5 Scalability

Chapter 3 presented some mechanisms that did not scale enough to be applied in the wide area, e.g. VP switching strategies. This problem appeared because the VPI field, whose maximum size is 12 bits, was used as group identifier. Thus, just the traffic from a limited number of groups may pass through a given link. Therefore, the new mechanism should cope with this problem by providing a larger group ID space than VP switching mechanisms.

5.2.6 Applicability to MPLS ATM LSRs

As explained before, one of our main concerns is to leverage the acquired knowledge in the ATM field by adapting it to the new technologies that may benefit from it. In particular, the new mechanism should take into account its potential application to MPLS environments.

Therefore, the implications of the design decisions over future implementations of ATM LSRs should be taken into account.

5.2.7 Simple implementation

Either in an ATM switch or in an MPLS ATM LSR, the new mechanism should have a simple implementation. State-of-the-art hardware just offers point-to-point or point-to-multipoint switching. The new mechanism should be able to provide multipoint-to-point and multipoint-to-multipoint communications. Thus, the simplicity requirement demanded to the implementation should become apparent in the form of minor modifications to state-of-the-art hardware to provide multipoint-to-multipoint capabilities. This means that, ideally, the processing at each of the modules in the switch (or LSR), like table management or cell switching, either should not be modified or modified as less as possible.

Furthermore, buffering strategies apart from modifying the traffic characteristics also introduce additional reassembly buffering that shall not be managed if cell-interleaving is allowed, thus simplifying the implementation twofold as far as buffering is concerned: first, reassembly must not be managed, and second, buffering is reduced.

5.3 Operation

Following the order in which the defined criteria were introduced, this section explains how each of them maps to the characteristics of the Compound VC (CVC) mechanism.

First, if group interactive applications must be supported with minimum changes to traffic characteristics, cells should be interleaved. So, CVC allows cell multiplexing, and thus, it must carry a multiplexing identifier somewhere in the cell.

Second, assigning PDU IDs contributes to a substantial saving of ID space, which could be used for other communications (unicast or multicast).

Third, this saving of ID space could not be complete if it was not coupled with a mechanism to negotiate the size of the identifier. In this way, just the right number of identifiers is used in a group.

Fourth, the use of PDU IDs instead of sender IDs jointly with the selection of the size of the identifier are fundamental design decisions that reduce the overhead introduced due to multiplexing management. But this additional overhead is completely eliminated if the multiplexing ID is carried in the fields of the header of the ATM cell. For CVC, the design decision was to carry the identifier in the VCI field. In this way, the VPI field may be used by carriers for traffic engineering purposes.

Consequently, the identification of the multicast group must also be carried in the VCI field, and thus the VCI field is divided into two parts. The utilization of the VCI field is depicted in Figure 11.

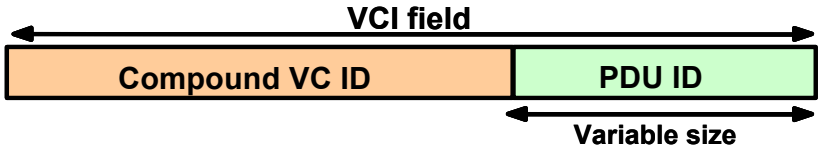

Figure 11. Compound VC (CVC).

The left part of the VCI field is called the compound VC identifier (CVCID) and is the identifier of the multicast group. Therefore, a multicast group uses several adjacent VCs to transmit its traffic. We call this set of adjacent VCs a *compound VC*, from where the name of the mechanism is derived. However, in CVC all these adjacent VCs are processed as a single (compound) connection.

The other part carries the PDU ID multiplexing identifier, which, apart from allowing the sharing of IDs and combined with the negotiation of the size, also permits a better usage of the bits of the ATM cell header.

The size of both parts is negotiated at connection establishment and they are locally mapped at each switch. As for the CVCID part, a static mapping is established at each switch along the path that is not changed during the connection. On the other hand, the PDU ID is dynamically modified every time the first cell of a new PDU arrives at the switch.

Finally, the utilization of part of the VCI field to carry the multiplexing identifier presents the additional advantage of benefiting from the header error correction mechanisms of ATM, unlike in other mechanisms, like SPAM or CRAM.

Back to the design criteria, and with respect to the fifth one (scalability), we measure it with two main parameters. First, the maximum number of groups that could send traffic through a given link, which is determined by the available group identifier space, and secondly, the maximum number of senders in any given group. Both parameters depend on the traffic characteristics of each particular scenario, but some general comments follow. If we take a look at the design choices, those strategies that either avoid cell-interleaving or that allow multiplexing inside a single VC are the most scalable of all native ATM multicasting mechanism with respect to the first parameter. Thus, CVC is not as scalable as them, but those mechanisms have other problems that are accrued due to the way they solve cell-interleaving, mainly delay characteristics alteration and extra overhead. On the other hand, VP switching is less scalable than CVC due to the utilization of the VPI as multicast group ID, which is not the case for CVC. Besides, the VCI space is larger than the VPI space and combined with the identifier size negotiation gives more flexibility in terms of the number of groups that may be supported. In addition to that, the sharing of IDs due to the utilization of packet IDs permits the utilization of few bits in the VCI for the packet ID, and thus, the remaining ones may be used as CVCID to identify the group.

With respect to the number of senders in the group, the packet ID allows to share a small identifier space among many senders. Though this sharing efficiency depends on the characteristics of the traffic sent by the sources, the evaluation work carried out in Chapter 6 shows that it may be significant. Additionally, the selection of the number of bits devoted to PDU identification allows to serve all the senders in the group, and thus provides what we could call a flexible scalability, because all the senders are served without wasting bits in unused overhead, which is the case for overdimensioned source ID strategies.

Furthermore, as the mapping in the CVC network nodes is local, no global source ID assignment is required and no central authority that could limit the scalability of the mechanism is required.

As for the applicability to MPLS, what has been said in section 2.3 allows us to think about the application of CVC in MPLS ATM LSRs as feasible. A further discussion on

how this application to MPLS would look like may be found in Chapter 7, where the architecture of an MPLS ATM LSR with CVC is discussed.

And finally, the implementation issues are equally dealt with in Chapter 7, but some general remarks may be introduced here. The dynamic management of PDU identifiers adds some complexity to current state-of-the-art hardware. But notice that this hardware does not provide multipoint-to-multipoint capabilities. Therefore, some additional processing is required. How to handle this processing is the main concern of the implementation issues of CVC. Two main options are considered in Chapter 7. In one of them, just slight modifications to current ATM hardware are required, thus contributing to the goal of having a simple implementation.

The implementations of other native ATM mechanisms also show additional complexity. For instance, SF requires some additional processing to manage the reassembly buffers, and token-passing schemes require a complex protocol to manage the token. The only exception to this is VP switching, which may be implemented in current ATM switches, if global source ID assignment is provided.

In light of what has been explained about the operation of CVC, it may be argued that CVC is a generic native ATM multicasting mechanism in the sense that it shares some characteristics with some of the previous mechanisms. For instance, it is like VP switching techniques in that the multicast connection is not a single VC, but multiple VCIs are assigned to the same connection. And CVC is similar to SPAM because cell multiplexing is allowed in the connection, but in this case not in a single VC but in a compound connection composed of several VCs. It is also generic because some previous mechanisms are particular cases of CVC. And the selection among them is done by means of the selection of the size of the PDU ID. As a consequence, they may also be applied in those cases in which they are appropriate. For instance, if the whole VCI is taken as PDU ID, CVC becomes DIDA. On the other hand, if four bits are defined as PDU ID, we have a mechanisms which behaves in the same way as WUGS subchannel mechanism. Besides, if the PDU ID is assigned no bits, and all the VCI is taken as CVCID, it corresponds to a point-to-point communication, which may be processed using the same forwarding table as multicast communications.

5.4 Traffic forwarding with CVC

As a final comment on the operation of CVC, this section introduces the aspects that arise in a scenario with multiple group members and switches. This will serve us to further explain some of the implications of the design criteria, not just in a single switch but in a complex scenario.

Figure 12 introduces a sample scenario where CVC is applied for a sample set of traffic parameters. It is homogeneous in the sense that all users in the multicast group (or CVC connection) receive the same treatment and introduce the same traffic to the network. A typical example of such a scenario is a fully interactive videoconference.

As the focus of this thesis is on forwarding, we assume that there are signaling and routing capabilities that allow the establishment of a shared bi-directional tree previous to the start of the forwarding of multicast data. These capabilities should allow the negotiation of the number of PDU IDs at each branch of the tree, which may be different in both directions. Figure 12 shows that the IDs chosen depend on the characteristics of the aggregated traffic. For instance, the link connecting a switch with a member of the group uses 1 ID in the ingress direction, because just this member uses it. But it uses 4 IDs in the egress direction because the aggregated traffic of the whole group is forwarded to all the components of the multicast group. And there are some other links where the aggregation of traffic does not give high traffic volumes, and thus, just two IDs are enough to multiplex the traffic through that link. Some hints on determining the number of IDs are given in Chapter 6.

We also assume that all switches in the network perform CVC forwarding. In this case, the operation is as follows. When a sender wants to transmit to the group, it sends its information through the single VC that connects it to the ingress switch. Once in the switch, if there is no additional traffic from other sources of the group being forwarded to the same output link, the packets will be transferred without PDU ID remapping, i.e. just the compound VC ID mapping part is required. For instance, when sender A in Figure 12 wants to transmit data, it sends its information through link A-S1 with the only ID assigned to this link for this multicast group. In switch S1, there is no information from other members of the group being forwarded from S1 to S2. Thus, PDU ID mapping in

S1 is not required and the packet is forwarded by just looking at the CVC ID switching table.

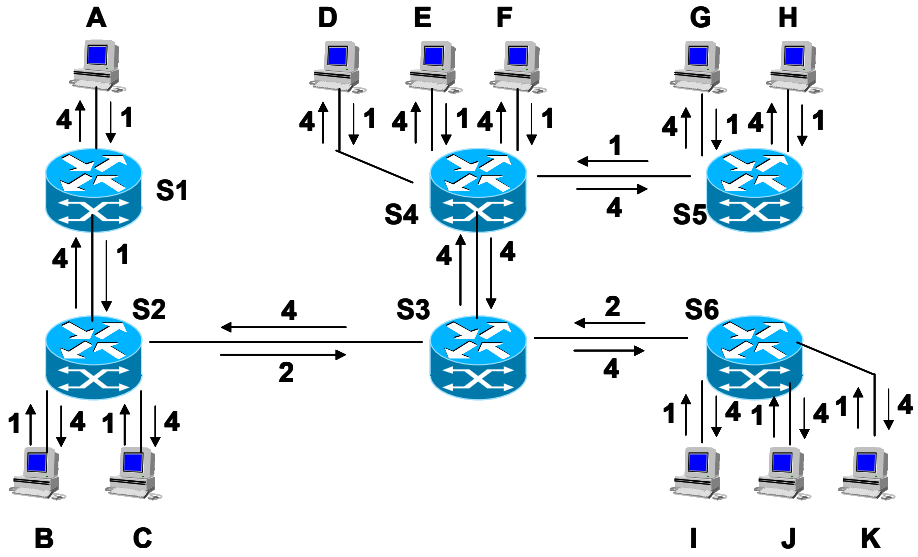

Figure 12. Example of multicast scenario with CVC

If there are other senders forwarding its information through the same switch, both flows will be aggregated. Therefore, the CVC forwarding mechanism will be used to map the PDU IDs. For instance, when the PDUs sent by sender A arrive at switch S2, they are multiplexed with those coming from senders B and C and they are forwarded from S2 to S3 through a 2-ID compound VC.

This example illustrates the benefits in the use of PDU IDs in the sense that, for example, in the direction from S2 to S3, three senders share two IDs, and in links where the aggregate carries the most traffic, the eleven members of the group may be served by just four IDs, less than half the number of identifiers we would require in source ID mechanisms. As a consequence of the operating characteristics, just the right number of bits in the header is used and there is more space to identify other groups, and in this sense, may scale better than other mechanisms like VP switching.

5.5 Classification

Having seen the operation of the new mechanism, we are able to include it in one of the classes presented in Table 4.

Table 6. Position of CVC in the native ATM multicasting mechanisms classification

Avoid cell-interleaving		VP switching		Allow multiplexing inside a VC			Multiple VC switching	
Buffering	Token control	Source ID	Packet ID	Added overhead	GFC	Signaling	Group ID	Packet ID
Cut-through (SEAM) CT-NC CT-T Store-and-Forward	SMART	VP switching VP-VC switching VP switched CLIMAX	DIDA	SPAM CRAM AAL5+ based CLIMAX	Subchannel (WUGS)	VC-merge scheduler	FMVC	DMVC SMVC **CVC**

CVC belongs to the Multiple VC switching group, and thus it shares the generic characteristics for these kinds of mechanisms, namely utilization of more than one VCI for the same multicast connection and allow multiplexing of cells inside a multiple VC connection (see section 3.5).

It belongs to the subtype that uses Packet IDs, as introduced in sections 5.2.2 and 5.3. The main drawbacks of the mechanisms that appeared in the literature prior to CVC were discussed in those sections. Therefore, CVC tries to improve them in those aspects that may benefit the most group interactive traffic, and additionally, tries to complete those aspects that were not fully defined and are common to all multiple VC switching mechanisms.

With respect to the former point, the buffering introduced in DMVC and SMVC mechanisms make they look like SF, and thus, apart from the additional management complexity of multiple IDs for the same connection, there is the management of the reassembly buffers in the same way SF did. Therefore, the benefit multiple VC switching mechanisms provide by allowing the multiplexing of cells may be reduced by this buffering. CVC tries to simplify this aspect and eliminates the reassembly buffering with the premise that if there is a correct dimensioning of IDs at each link, this block is not

required. As a consequence, a simpler implementation is attained and the variation of the delay characteristics that reassembly buffering introduces is eliminated.

With respect to the negotiation aspects, and this leads us to the latter point, [Venkateswaran 98] does not discuss how the number of identifiers is determined. This thesis tries to go a step further and Chapter 6 discusses this issue.

5.6 Example of operation. Comparison with other mechanisms.

After the classification, and for further clarifying the operation of CVC, the last sections of this chapter are devoted to make a comparison of the new mechanism with the most representative ones that were previously introduced in the literature. A graphical comparison of the behavior of some of these mechanisms is presented in Figure 13.

We have chosen SF and CVC as the most representative ones because avoiding cell interleaving using token control leads to highly complex management and usage of many VCs. The mechanisms based on the assignment of identifiers (VPI or VCI) per source suffer from scalability problems when a large number of groups have to be maintained. However, they are also represented in the figure for comparison purposes. The mechanisms that allow multiplexing within a VC add extra overhead (modifying AAL5 PDU or using RM cells), or are limited to the UNI (using the GFC field of the ATM cell). For these reasons, the most interesting comparison is between SF, because it is the approach the most likely to be implemented in ATM switches and MPLS ATM-LSRs, and CVC, which in turn includes other mechanisms, as explained in section 5.3.

In this example, there are four incoming PDUs (A through D). The upper part (labeled as incoming cells) represents the time arrival instants of the cells belonging to each PDU. For instance, PDU A is composed of 4 cells arriving with a timing of one every four time units, where one time unit corresponds to the time it takes to transmit one cell (or Cell Transmission Unit, CTU). Higher time values are represented to the left of the figure, in this way, cells that first arrive at the switch are represented to the right.

The part of the figure labeled as *outgoing cells* presents the exit instants of the *incoming cells* for each mechanism after having passed through processing in the switch. Each horizontal arrow (jointly with the cells above it) represent the exit instants for the

mechanism presented to the right of the arrow. The number in parentheses beside the name of the mechanism represents either the number of IDs for CVC and Source ID mechanisms or the number of reassembly buffers for store-and-forward (SF) buffering techniques.

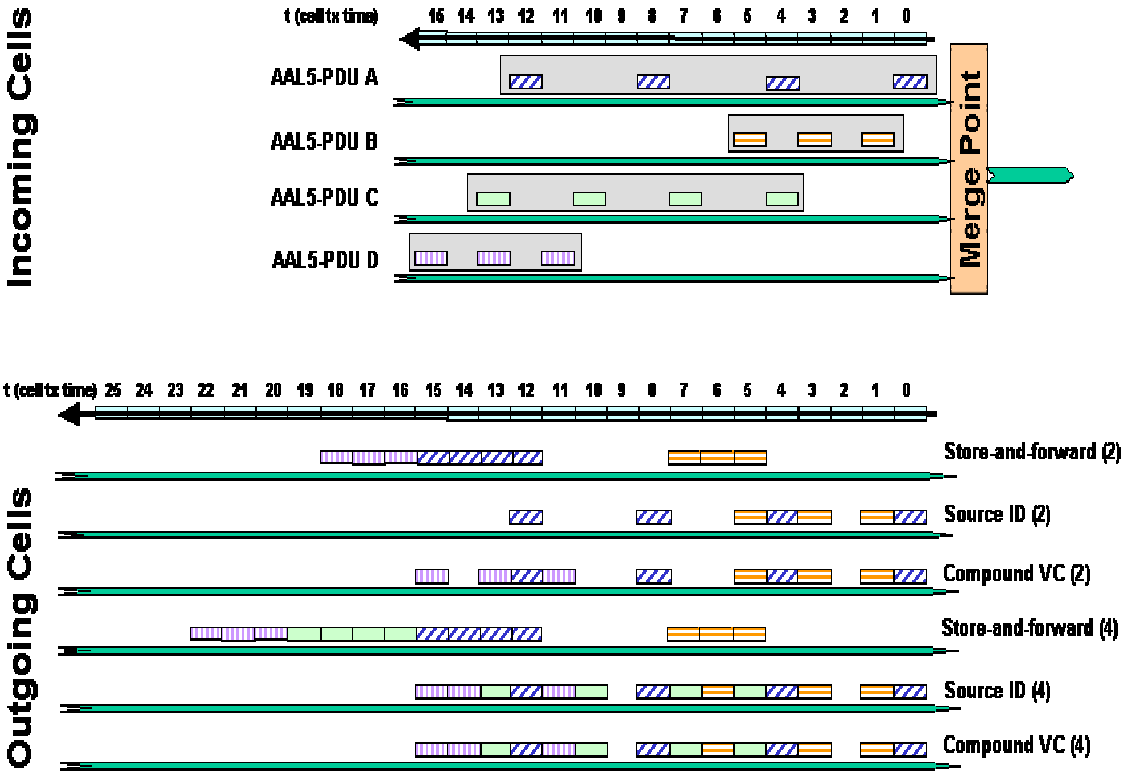

Figure 13. Example of operation of CVC compared to other native ATM multicasting mechanisms

According to the operation of the mechanisms, the following issues should be noticed:

If the first cell of a PDU is discarded, so are the remaining cells, as in Early Packet Discarding strategies.

For comparison purposes only, we have chosen 2 and 4 reassembly buffers for SF. It does not claim to be a general case in practice.

The initial state of the system corresponds to empty buffers.

For the sake of clarity, we have represented no delay for the processing time of the switch in all cases. The only delay introduced is due to the buffering in the reassembly process or to the one due to asynchronous multiplexing of ATM.

To simplify the figure, the PCR of the output link is taken to be the link capacity. Thus, in a burst, cells are transmitted at contiguous cell transmission times.

In Store-and-Forward(2), the first cell to get to the merge point (the switch) is that of PDU A. Once it gets there, it starts to fill one of the two reassembly buffers. At t=1, the first cell of PDU B arrives at the merge point and starts to fill the second (and last) buffer available. Therefore, at t=4, PDU C can not get a free buffer, and thus, all the cells of that PDU will be discarded as they arrive to the switch. At t=5, the last cell of PDU B arrives and frees one of the buffers because all its cells are moved to the output transmission buffer in an atomic manner and are transmitted through the output link at time instants 5, 6, and 7. But this buffer is not occupied by the following cells of PDU C because it has been marked to be discarded. On the other hand, at t=11, the first cell of PDU D starts to fill this free buffer. At t=12, the last cell of PDU A arrives and, as a consequence, all the cells are moved to the output transmission buffer and transmitted at time instants 12, 13, 14, and 15. The reassembly buffer is also freed. Finally, at t=15, the last cell of PDU D arrives, and all its cells are processed in the same form. But in this case, these cells must wait one time unit to be transmitted because the last cell of PDU A occupies the resources. Therefore, they are transmitted at time instants 16, 17, and 18.

As for Source ID(2), we assumed that each of the PDUs came from a different sender. Therefore, at t=0 PDU A allocates the first ID and at t=1 PDU B allocates the second ID. For the rest of the transmission to the group, just cells coming from the senders that generated these two PDUs are accepted, and the rest are discarded because there are no spare IDs.

Now, if we consider the output traffic pattern for CVC(2), we see that, as cells are allowed to be multiplexed, the cells are transmitted to the output port as they arrive in case there are enough identifiers. For instance, at t=0 and t=1, PDUs A and B get one ID each. On the other hand, at t=4, PDU C can not get a free ID and is marked to be discarded. At t=5, the last cell of PDU B arrives and frees one ID, which is reused by

PDU D. This shows the advantages of using PDU IDs as opposed to Source IDs. Finally, at t=12, another ID is freed as the last cell of PDU A arrives, and the other one is freed when PDU D ends (t=15).

If the traffic patterns obtained for each of the strategies are compared, it may be observed that there is a remarkable difference in the way SF transmits the cells and the way it is done by source ID and CVC, which allow the multiplexing of cells. If there is no delay introduced in the processing, which is the assumption we made, the ideal case, would be that cells are transmitted at the same time instant as they arrive, and the only delay they may experience is due to another cell that arrives at the same time at the merge point, i.e. due to the asynchronous statistical multiplexing of ATM. Actually, this is the case for Source ID(2) and CVC(2), but it is not the same for SF(2). Therefore, if we do some back-of-the-envelope calculations for the delay (in cell time units) for SF, the average delay due to reassembly buffering is 5.1 cell time units. This result is obtained by subtracting the time a given cell is transmitted through the output port and the time it arrived to the merge point. Only cells that got an ID are considered for this calculation, that is, 10 cells. Table 7 illustrates the process.

Table 7. Cell delay calculations for Store-and-forward(2) in Figure 13.

PDU A		**PDU B**		**PDU D**	
Cell	Delay (CTU)	Cell	Delay (CTU)	Cell	Delay (CTU)
1st	12-0=12	1st	5-1=4	1st	16-11=5
2nd	13-4=9	2nd	6-3=3	2nd	17-13=4
3rd	14-8=6	3rd	7-5=2	3rd	18-15=3
4th	15-12=3				

And the average delay is calculated as follows,

$$\text{Average delay} = \frac{12+9+6+3+4+3+2+5+4+3}{10} = 5.1 \text{ CTUs}$$

On the other hand, the ideal case (source ID and CVC) gives an average delay of 0 cell time units, because in this case no two cells try to leave the switch at the same instant, i.e.

there is no delay introduced by statistical multiplexing. Furthermore, the burstiness of the output traffic for SF is higher than in the other mechanisms.

The same comments apply for SF(4), Source ID(4), and CVC(4). Nevertheless, the throughput obtained is higher because there are more identifiers or reassembly buffers available, and thus, less PDUs are lost. However, the more PDUs are multiplexed, the more the outgoing traffic pattern is modified. For instance, if we take PDU D in SF(4), we see that is has to wait more CTUs than in the SF(2) case because there are more cells in the output transmission buffer before its first cell may be transmitted. The average delay calculations for SF(4) will reflect this increase in delay. They are presented in Table 8.

Table 8. Cell delay calculations for Store-and-forward(4) in Figure 13.

PDU A		**PDU B**		**PDU C**		**PDU D**	
Cell	Delay (CTU)	Cell	Delay (CTU)	Cell	Delay (CTU)	Cell	Delay (CTU)
1st	12-0=12	1st	5-1=4	1st	16-4=12	1st	20-11=9
2nd	13-4=9	2nd	6-3=3	2nd	17-7=10	2nd	21-13=8
3rd	14-8=6	3rd	7-5=2	3rd	18-10=8	3rd	22-15=7
4th	15-12=3			4th	19-13=6		

And thus, the average delay is,

$$\text{Average delay} = \frac{12+9+6+3+4+3+2+12+10+8+6+9+8+7}{14} = 7.07 \text{ CTUs}$$

On the other hand, the average delay for Source ID(4) and CVC(4) is 0.21 CTUs in both cases, because, unlike in the 2 ID case, there is some delay due to asynchronous multiplexing of cells. There is one CTU of delay for the 3rd cell of PDU B, the first one of PDU C, and the second one of PDU D. The rest of cells exit the switch at the same time instants they arrive, therefore, its delay is 0 CTUs, and thus, following the same steps as before, 3/14=0.21 CTUs.

5.7 Additional operational aspects of CVC

This section describes some of the lateral aspects of the operation of CVC. They are lateral to this thesis in the sense that they are not the focus of it, but this does not mean at all they are unimportant. What follows is some thoughts on how signaling and interoperability with current switches may be dealt with. However, they are not treated in depth as the goal is just to provide some initial guidelines on how each of this problems may be solved and not to provide a complete solution.

5.7.1 Signaling

Some possible solutions to the signaling aspects of CVC were presented in [Mangues 00b]. The following paragraphs are devoted to explain them.

The PDU ID size negotiation will take place at connection establishment. How this size is determined is the concern of the following chapter. Before entering these aspects, it could be interesting to see the different possibilities for establishing a CVC connection.

The establishment of the connection, with the consequent creation of the group, could be handled in a similar way as that of SEAM [Grossglauser 97], which uses core-based trees. Member-initiated joins are supported for scalability reasons. The join procedure could be similar to Leaf Initiated Joins (LIJ) defined in UNI 4.0, not just for receivers, but also for senders.

Core-initiated joins are also considered. This latter approach would be interesting when quick establishment of a group communication initiated by a central coordinator is required.

Apart from all these previous considerations, some distinctive features appear in the signaling of CVC due to having to consider compound VCs instead of normal VCs. Therefore, the signaling at the network to network interface (NNI) must be modified to treat a set of VCs as a group.

For the UNI, there are three options. The first one consists of designing an extension of UNI signaling for CVC. It would consist of modifying current messages by introducing elements that consider the compound VC characteristics as a whole instead of those of

each individual VC inside the CVC connection. For instance, there would be a joint traffic descriptor.

Another option would be to establish, by using standard UNI, as much VCs as IDs the CVC connection requires. In this case, a higher level entity would be responsible for managing the information coming from these individually established VCs, and to consider them as a whole. Besides, the egress switch will distribute the information flow among these VCs.

The third one would be to keep standard UNI signaling. That is, the end-user would receive all the information from the group through a single VC. If this solution were adopted, the egress switch would be in charge of avoiding cell interleaving by applying a buffering technique at UNI interfaces, as suggested in [Calvignac 97]. Traffic characteristics will be modified. However, it may be acceptable because buffering is just carried out at the egress switch.

In a generic scenario, the number of IDs at each link connecting two switches could be different in each direction depending on the aggregate traffic from the sources. However, signaling could be simplified by considering the same number of IDs in both directions, but this would lead to ID space being wasted.

5.7.2 Interoperability

A typical scenario to study interoperability would be that composed of islands of CVC switches and islands of current equipment. The most reasonable solution in this case seems to be that proposed in [Turner 97] where the egress switch of the CVC island establishes as many VCs as multiplexing IDs are assigned to the group (N). Conventional signaling will be used to establish these circuits, but the egress switch will schedule the traffic going through each VC, so as to share all the VCs among all the senders. Therefore, N conventional VCs would be used to connect non-CVC islands.

Another option to reduce the signaling overhead at end-systems consists of implementing VC merging at egress switch. This would allow the end-system to receive all the traffic in the group by establishing just one or very small number of VCs. On the

other hand, the switch would be more complex and the delay, CDV, and burstiness could be slightly increased.

Ingress switches to the CVC island should be in charge of transforming conventional VCs into CVC connections by maintaining a special table that would map a VC to a group connection. Mapping would assign a free ID to each PDU just as it does in normal CVC operation. A previous setup phase of the CVC connection is required from the ingress switch to the rest of the switches.

Chapter 6

EVALUATION OF COMPOUND VC

This chapter presents the evaluation of some of the CVC mechanism. Comparisons with other native ATM multicast mechanisms are also provided. These comparisons have been carried out in two main aspects, namely throughput and ID dimensioning. The former shows that the throughput obtained is the same as in the store-and-forward mechanism and the advantages of multiplexing per PDU. The latter is further developed while trying to solve the ID dimensioning problem at connection establishment. These issues are mainly studied through simulation, though a theoretical approach is also given to confirm the validity of the results obtained in the scenarios under study.

6.1 Introduction

This chapter deals with the quantitative evaluation of the Compound VC mechanism. The evaluation focuses on parameters that help to demonstrate the advantages claimed by CVC in previous chapters with respect to other native ATM multicasting mechanisms. In this sense, some of the following sections compare the behavior of CVC with other proposals with respect to a given parameter. As Store-and-Forward (SF) seems to receive particular attention, most comparisons are made with this mechanism. The parameters under test are mainly throughput and the number of required identifiers.

Throughout this chapter, SCR and average per source are used interchangeably as well as PDU length and number of cells per PDU.

This chapter is organized as follows. The following section deals with the evaluation of throughput. The simulation environment used for this evaluation is also introduced. Next, the focus is on the dimensioning of the number of IDs at connection establishment. A theoretical analysis of the behavior of the sources is complemented with simulations. Results for various scenarios as well as for various parameters are presented and discussed.

6.2 Evaluation of throughput

This section is devoted to evaluate the throughput obtained when CVC is implemented in the switch given some traffic characteristics at the sources. Therefore, as a first step, it may be convenient to define the parameter we are evaluating. For the purposes of this evaluation, the term throughput is understood as the maximum output rate that the chosen number of multiplexing identifiers allows. That is, in this section, we assume we never run out of any resource except the number of multiplexing identifiers. For this reason, we simulate an infinite output buffer in the switch, and thus, the only cell losses are due to the lack of free PDUIDs when the first cell of a packet arrives at the output port of the switch we are stressing. Part of these results were presented in [Mangues 99].

6.2.1 Simulation scenario

The simulated scenario (Figure 14) consists of some sources sending traffic to the same multicast connection and traversing the same output port of the switch. The sources are homogeneous, i.e. they have the same statistic (Poisson) for the arrival process with the same mean interarrival time, which follows an exponential distribution. The length of the PDUs also follows a geometric distribution for all sources. An ON-OFF model models each source (see Figure 15). Once in the ON state, the source transmits cells at its peak cell rate (PCR). Besides, each source just sends one PDU at a time, i.e. cells belonging to one PDU are sent first, and then, cells belonging to the following one are sent. The number of sources is varied to compare the behavior of the mechanisms with different input traffic loads. For the results presented in next section, the output PCR for the group doubles that of each source. And each source introduces $0.2*PCR_{in}$ of traffic load on the average. The concrete values used in the simulation follow. The number of sources ranges from 1 to 10, with each source sending one cell every 10 cell transmission units (CTUs) at most, which leads to a $PCR_{in} \approx 15$Mbps for an STM-1 link. As for the average, the sustainable cell rate (SCR) is approximately 3Mbps (0.2*15Mbps). At the output, one cell is sent every 5 CTUs, thus resulting in a PCR of the output link (PCR_{out}) of approximately 30Mbps. Finally, the mean PDU length is 5 cells.

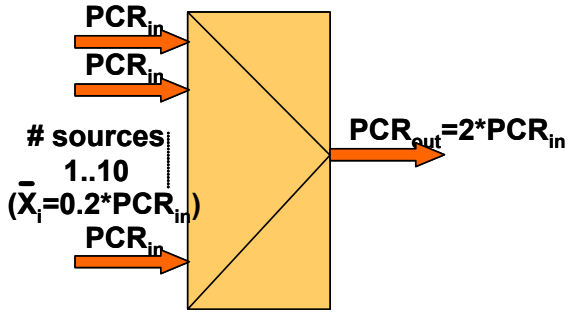

Figure 14. Evaluation of throughput. Simulation scenario.

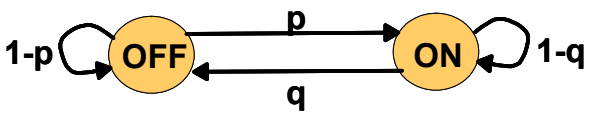

Figure 15. Evaluation of throughput. Source model.

The switch implements three multicasting strategies, namely Store-and-Forward (SF), Source ID, and CVC. For our purposes, the most important parameter that characterizes how the multicast connection is processed in the switch is the number of identifiers (or reassembly buffers for SF).

Some previous assumptions are required to be able to make the comparison between mechanisms that use multiplexing IDs and Store-and-Forward (SF). This latter mechanism does not use ID assignment, as there is no cell interleaving. What it does assign is reassembly buffers. Therefore, for the purposes of our comparison, the number of identifiers is matched with the number of reassembly buffers in the output port of the switch we are studying. Though this comparison does not strictly represent a general case, we think it is reasonable for the following reasons. First, finding a free reassembly buffer may be compared to finding a free multiplexing ID, as both are assigned to a given PDU when the first cell arrives and the association between ID/buffer and PDU is kept until the last cell of the PDU arrives. However, the way output buffers are managed at the output port is not the same as the way IDs are managed in CVC or Source ID strategies. In the first case, there is usually a pool of memory devoted to reassembly buffers that is shared by all the SF connections through that port. On the other hand, IDs are used by just one multicast connection. Notwithstanding this, we still think that comparing the same number of reassembly buffers in SF as IDs in the other mechanisms is a reasonable choice in the sense that as reassembly buffers are shared by many connections, approximately the same number of IDs as reassembly buffers will be used on the average by a single connection in case the output port is stressed, as is the case in throughput measurements.

As for Source ID strategies, they can only offer service to a number of sources not greater than the number of IDs. Therefore, we assume that, at the beginning of the simulation, each ID is assigned to a source, and this binding lasts until the end of the simulation.

6.2.2 Results

Prior to entering in the detailed discussion of the results, a comment on the interpretation of the figures may be in order. In the following figures, both X and Y axis are normalized.

The former is normalized to the PCR of the sources (PCR_{in}) and the latter is normalized to the output PCR (PCR_{out}). Therefore, a value of 2 in the X axis means that the average rate of the aggregated traffic coming from the sources is twice the PCR of one source. Recall from the previous section that each source introduces $0.2*PCR_{in}$ on the average. Therefore, the aggregated traffic introduced by 10 sources corresponds to a value of 2 in the X axis. As for the Y axis, a value of 1 corresponds to the output capacity assigned to the multicast connection being completely full, i.e. the average aggregated rate equals the maximum rate allowed to this connection, that is, PCR_{out}. Therefore, the ideal curve for throughput is in fact a line that means that all the cells in the input traffic flow are allocated an ID or reassembly buffer, and thus, there is no loss due to lack of IDs, and as a consequence, no cell is discarded. For the simulation parameters chosen, and if normalizations are taken into account, the expression for this line is:

$$\text{Throughput} = 0.5 * \text{input traffic} \quad (1)$$

Figure 16 presents the comparison of the throughput obtained for the three mechanisms under comparison (SF, Source ID, and CVC) with 2 IDs (or reassembly buffers). It shows what could be expected after examining the behavior of the mechanisms presented in Figure 13. The results for the three methods show no significant differences with aggregated input loads lower than 40% of the PCR of one source. To explain why the results are the same we should take a look at how the parameters of the simulation are chosen. According to what has been explained in the previous section, two sources produce a load of $0.4*PCR_{in}$. Therefore, in the Source ID mechanism, two IDs are enough to process all the traffic arriving to the group. And the same happens with 2 PDUIDs for CVC and 2 reassembly buffers for SF. There are just two sources sending, and, as explained above, each source does not interleave cells belonging to different PDUs. But these low loads will not be very common even in local environments as one of the main applications of multicast is multimedia communications. Such communications are characterized by high bandwidth consumption, essentially due to video transmission.

For Source ID, the throughput is limited to the traffic generated by the sources that allocated an ID at the beginning of the simulation. That is, once the number of sources reaches the number of available IDs for a group, the throughput stops increasing and the

traffic from the rest of the sources is discarded, as there are no free IDs. That is the reason why this mechanism presents a constant throughput for input loads greater than 0.4, which is the load imposed by 2 sources. Usually, the mechanisms using this philosophy overcome this drawback by overdimensioning the number of available IDs. See section 5.2 for further discussion on the drawbacks of such approach.

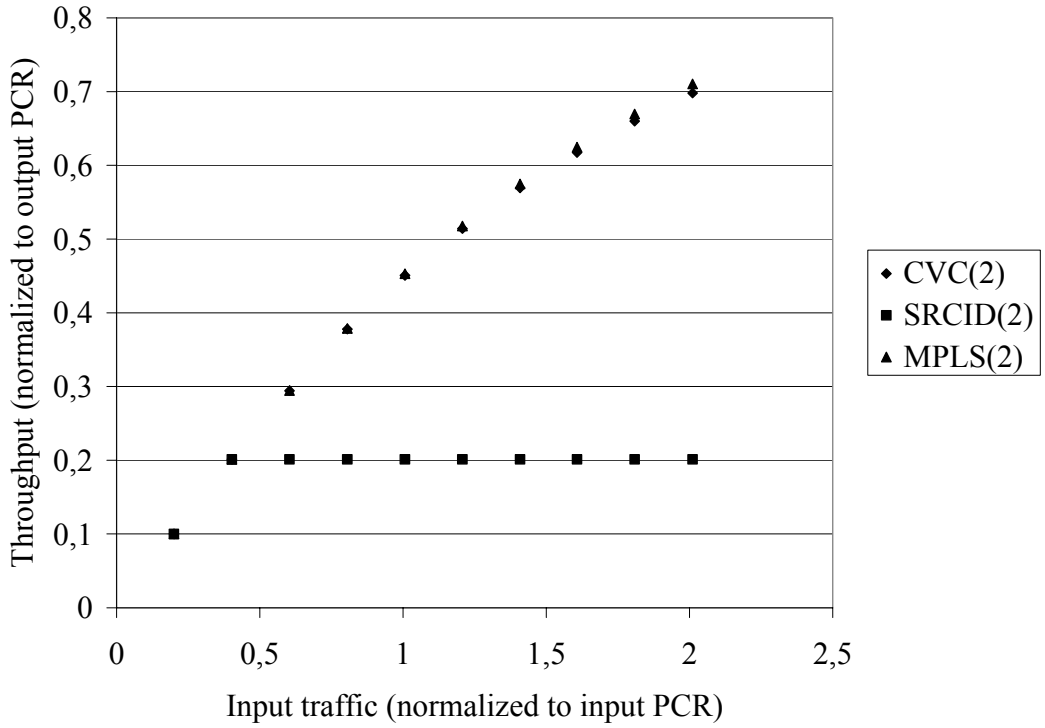

Figure 16. Comparison of SF(2), Source ID(2), and CVC(2)

The throughput obtained for SF and CVC presented no significant differences, which may also be guessed from the qualitative comparison in Figure 13. This result seems logical according to what is explained above in the sense that assigning a reassembly buffer to a new PDU arriving at the output port of the switch may be seen as an equivalent operation to assigning a PDUID to this same PDU.

We may also see in this figure that with two IDs the throughput obtained is closer to the ideal case just for low volumes of input traffic. A more detailed study of the curve of throughput with respect to the ideal one when the number of IDs is varied is presented in next figure.

A comparison of the throughput obtained with different number of IDs with CVC is presented in Figure 17. It may be observed that with few IDs, the throughput obtained is high even with a number of IDs much lower than the number of sources. Even with 1 ID, the throughput obtained with a global input load of 2.0 is remarkable (0.4). When the number of IDs is increased, there is a substantial gain in the behavior of CVC, because throughputs up to 0.7 are obtained for 2 IDs. And with 4 IDs, the behavior is very close to the ideal case. The throughput obtained in this case for 10 sources is 0.97. No difference between the 8 IDs case and the ideal case is shown. That is, for this traffic, with just 4 IDs the traffic characteristics are respected and the behavior is approximately that of the ideal case.

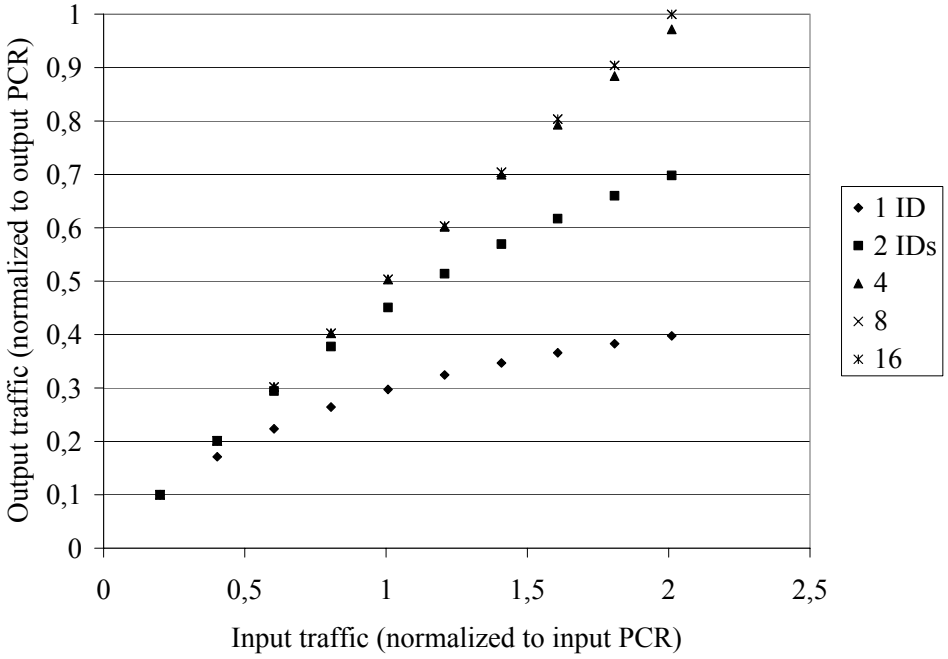

Figure 17. Throughput for CVC with different number of IDs

These results lead us to conclude that CVC presents good throughput characteristics, at least when compared to SF. We may also observe that both strategies present much better results than Source ID strategies. The only way these latter proposals may offer equivalent levels of throughput is by overdimensioning the ID size, and thus, by introducing extra overhead. An additional result is that with a low number of IDs, high throughputs may be obtained, and as a consequence, the VCI space consumption of CVC

may be minimized. Besides, the throughput curves also serve us to notice the advantages in terms of ID sharing when PDUID is used instead of Source ID.

6.3 Evaluation of the number of identifiers

The previous section served to study the throughput behavior of CVC and to compare it with that of other mechanisms. We have seen that high throughputs may be obtained with a small number of identifiers if they are assigned to PDUs instead of sources. However, in CVC, this PDUIDs are carried in the VCI field of the ATM cell header (see Chapter 5), and thus, we would like to be precise in the calculations of the number of IDs required in the sense that most traffic in the group may be served, while VCI space consumption is minimized. In this way, more VCI space is left for other multicast groups and the scalability of the mechanism is increased. Therefore, one of the main problems to solve for CVC is the dimensioning of such variable-length ID.

The dimension of the PDUID for a given multicast group at the output port of a given CVC switch should be chosen according to the traffic characteristics of the group and the group size. The characteristics of the aggregated traffic observed by a switch are ultimately related with the traffic characteristics of each particular sender. The traffic from the sources may undergo many multiplexing stages along the shared multicast tree. Furthermore, in the general case, sources may send traffic to the group with traffic parameters that are very different from those of other sources. All the above makes the characterization and study of an aggregated traffic flow very difficult in the general case. This complexity hinders the generation of valuable conclusions. For these reasons, we focus on a simple scenario, which is described below. However, as simple as it is, this scenario will allow to derive useful conclusions with respect to PDUID strategies, and particularly, CVC. In this scenario, simulated parameters are more under control than in the general case, and thus, the dependencies between traffic parameters may be more easily studied.

In this section we try to determine the dependence of the required number of IDs as a function of the parameters of one source. This makes sense because we will study a homogeneous environment, i.e. all the senders transmit traffic with the same

characteristics. The source parameters considered in this initial study are the PCR, the SCR, and the average number of cells per PDU (or PDU length).

With respect to group characteristics, we see that passive members of the group, i.e. hosts that only act as receivers, do not have any effect on the traffic in the tree. As a consequence, the group size is characterized by the number of senders (N) in the group, when the purpose is to dimension the ID.

6.3.1 Methodology

To attain the dimensioning goal, we will study the PDU losses produced due to running out of identifiers in a given switch and its dependence on the above parameters. The methodology we follow starts by calculating the histogram representing the frequency of the number of slots in which a given number of simultaneous PDUs is being transmitted through a given output port, i.e. the probability mass function (pmf) of the number of simultaneous PDUs. The result of this first step is a graph like the one represented in Figure 18. It may be observed that in the graph of the example, the most frequent number of simultaneous PDUs is three, with a percentage value around 0.22. That is, in the 22% of slots, 3 PDUs were simultaneously traversing the switch in this simulation.

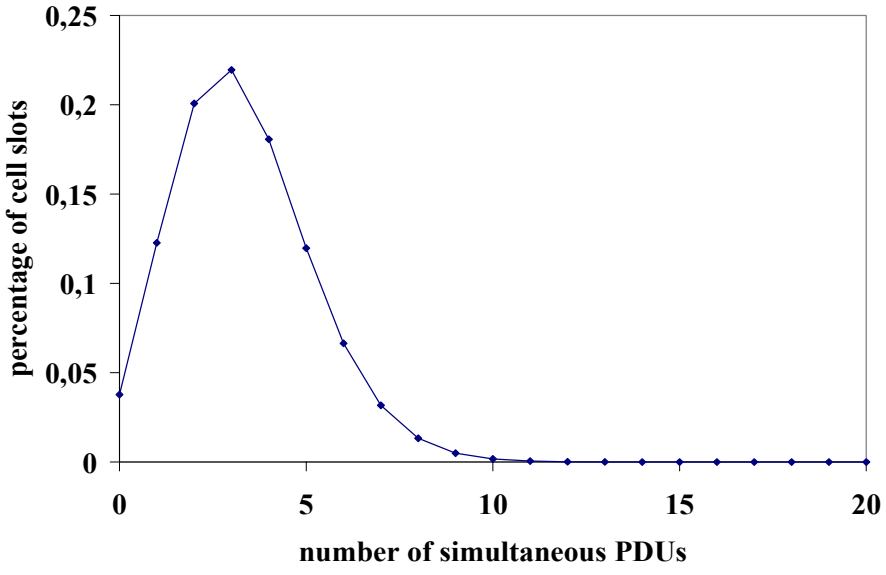

Figure 18. First step. Probability mass function of the number of simultaneous PDUs.

The second step consists in calculating a graph for the PDU loss probability (PLP) due to running out of identifiers, i.e. the graph represents the probability of loss of an arriving PDU as a function of the number of IDs (nID) used (see Figure 19). That is, given a value of nID, the corresponding probability of losing an arriving PDU is calculated from the previous one by adding the probability mass function values from nID up to N-1 simultaneous PDUs. The addition starts at nID because when a PDU arrives and there are already nID simultaneous PDUs using the IDs, the arriving PDU will be lost. And the addition ends with the term N-1 and not N, because, if we have N sources and we have one newly arriving PDU, it must come from the remaining source that is not sending (recall that any source just sends one PDU at any given time). Therefore, to calculate PLP as a function of nID, the previous addition is carried out for nID values from 0 to N to obtain a graph like that in Figure 19. Notice that when nID=0, the probability of losing the PDU is 1, because as there are no IDs assigned to the connection, no PDU can get an ID. On the other hand, when nID=N, this probability is 0, because we are assigning an ID for each source, and as each source transmits just one PDU at any given time, no traffic is lost.

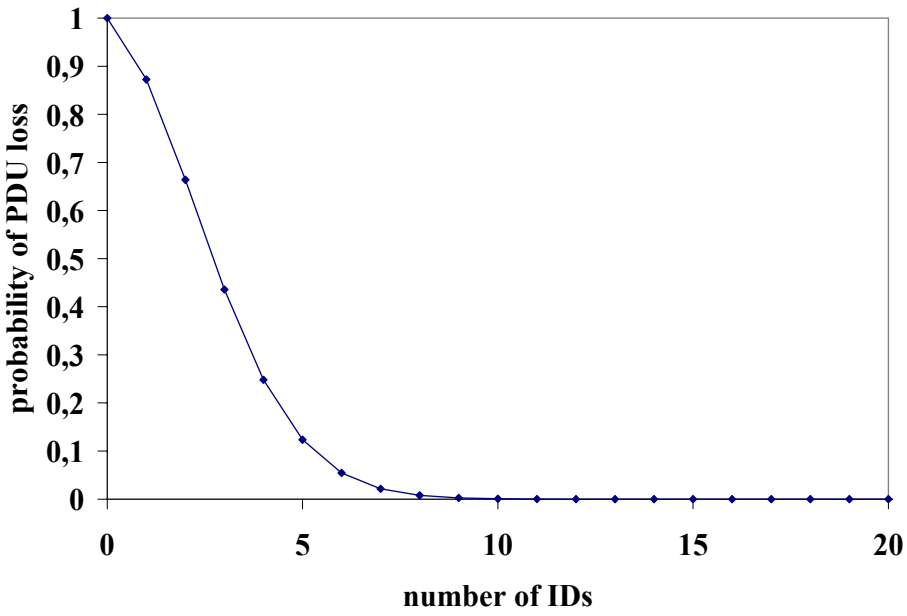

Figure 19. Second step. PDU loss probability as a function of the number of IDs in the multicast connection.

And finally, the probability that a PDU passes through a switch may also be obtained from the probability mass function above. In this case, we add the values ranging from 0 to nID-1 simultaneous PDUs. That is, this third graph gives the opposite probability to that calculated in the second step. How such a curve may look like is represented in Figure 20. In this graph, a value of one corresponds to a situation in which all PDUs that are sent by the sources are served by the switch and are not lost due to lack of multiplexing IDs.

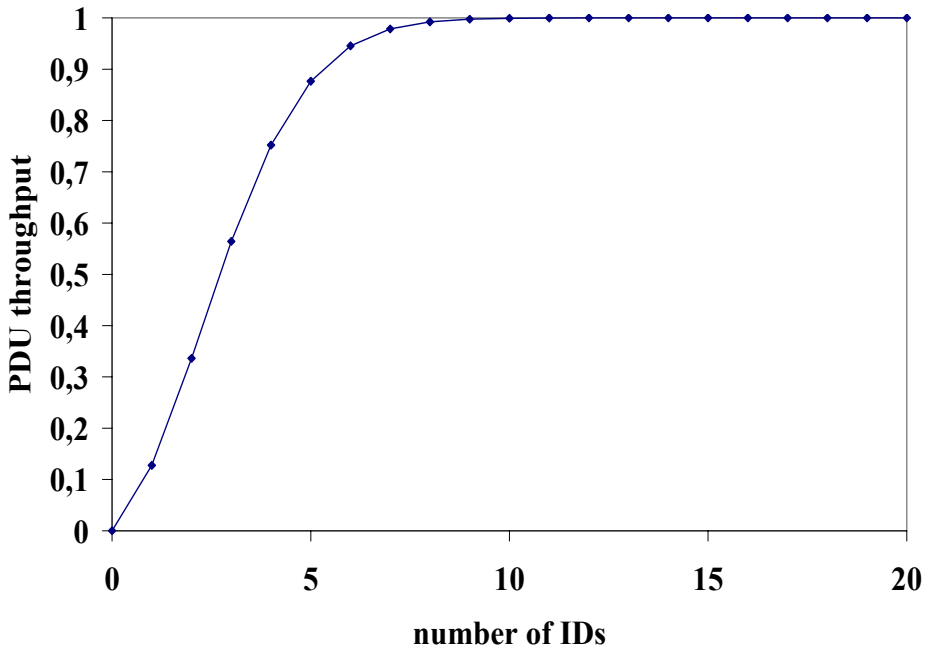

Figure 20. Third step. Probability that an arriving PDU is assigned a free ID as a function of the number of IDs in the multicast connection.

As one of the goals of the section is to obtain a value for nID as a function of traffic and group characteristics, our study will mainly focus on the PDU loss probability graph. The idea is to find some dimensioning rules that relate such probability with the parameters under consideration. The final goal of such rules would be to allow the dimensioning of the PDUID field at CVC connection establishment given a PDU loss probability acceptable to the user. Furthermore, if the obtained PLP value is lower than that caused by buffer overflow at switches or any other kind of transmission losses, the multiplexing process would not be the limiting factor of the multicast communication.

The dimensioning problem is tackled from three different approaches, which respectively correspond to the following three sections. In the first one, a probabilistic expression is used to calculate the PLP function. The second approach is based on the results obtained through simulation, which are compared with the previous ones. An finally, the environment under test is evaluated by using Erlang-B calculations. All of them seem to provide quite similar results for our purposes, and thus, all these approaches serve to validate the results obtained through simulation and to provide a way for the dimensioning of CVC communications.

6.3.2 Theoretical approach

Apart from the practical approach to the dimensioning problem through simulation, a theoretical approach towards our goal may be based on an expression that appears in the literature. An analytical expression for finding the probability that a burst of the GFC mechanism is lost is presented in [Turner 97] (see section 3.4.2 for an explanation of the operation of Turner's mechanism).

$$\sum_{i=h}^{n-1}\binom{n-1}{i}p^{i}(1-p)^{(n-1)-i} \qquad (2)$$

Following Turner's notation, this equation depends on the number of sources (n), the number of subchannels (h), and p is the probability that any given source is transmitting a burst, which, in turn, depends on the average number of busy sources (m). Therefore, p is equal to m/n. When applied to CVC, a burst is taken to be a PDU.

However, parameter m is too generic, and thus not always available, to be useful at connection establishment when trying to determine the number of required IDs. Other more easily obtained parameters, which could be directly related to the source, should be used for this purpose.

With this goal in mind, let us characterize the behavior of the sources. The scenario under study in our case consists of some bursty sources sending traffic to the same output port of a switch. They are characterized by their average cell rate (SCR), their peak cell rate (PCR), and the length of the burst in cells (B), which is related with the number of cells per PDU as explained below. The sources are homogeneous, i.e. they are all

modeled by means of the same statistics with the same parameters. An MMDP (Markov Modulated Deterministic Process) with one active state and one silence state is used to model each source. In fact, this corresponds to a particular model of an ON-OFF source (Figure 21). For this model, the sojourn time at both states follows a geometric distribution. At OFF state, the source does not send any cells. At ON state, it sends cells at its peak cell rate (see Figure 22).

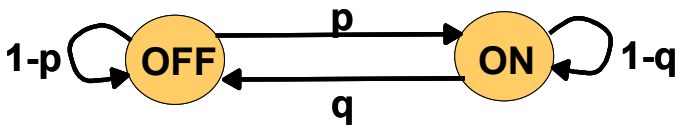

Figure 21. ON-OFF source model

The relationship between the average length of the burst (B) and the average number of cells per PDU may be derived with the help of Figure 22.

$$B = (\text{PDU\_length(cells)} - 1) \cdot \frac{\text{Link\_capacity}}{\text{PCR}} + 1 \qquad \text{(cells)} \tag{3}$$

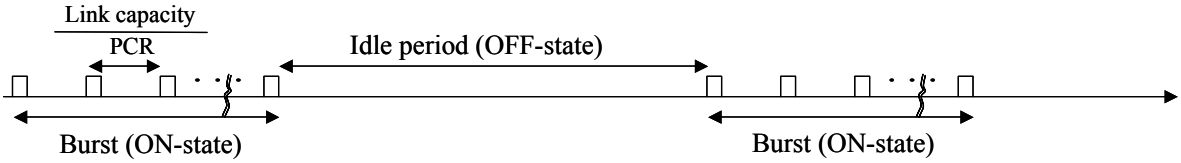

Figure 22. Behavior of the ON-OFF source

It is a Markov chain in the sense that both the states and the times at which a state transition may take place are discrete. The analysis of this Markov chain is required in order to obtain the probabilities of being at ON and OFF states. Therefore, the state transition matrix is:

$$\begin{bmatrix} 1\text{-}p & p \\ q & 1\text{-}q \end{bmatrix} \tag{4}$$

and the system to solve is:

$$[P_{ON} \;\; P_{OFF}] = [P_{ON} \; P_{OFF}] \begin{bmatrix} 1\text{-}p & p \\ q & 1\text{-}q \end{bmatrix} \tag{5}$$

from where we obtain that the probabilities of being at the ON and OFF states are:

$$P_{ON}=\frac{p}{p+q} \qquad P_{OFF}=\frac{q}{p+q} \tag{6}$$

and the transition probabilities are related to the source parameters as follows [Bolla 96]:

$$p=\frac{1}{B(b-1)} \qquad q=\frac{1}{B} \tag{7}$$

where b=PCR/SCR is the burstiness of the source. It is interesting to note that the transition probabilities do not directly depend on the peak cell rate or the average cell rate of the source, but on its ratio, the burstiness. And the other involved parameter is the length of the burst. Furthermore, if the p and q expressions deduced by [Bolla 96] are used in the P_{ON} and P_{OFF} expressions, we obtain:

$$P_{ON}=\frac{1}{b} \qquad P_{OFF}=\frac{b-1}{b} \tag{8}$$

which just depend on the burstiness.

With respect to the average sojourn time at ON state, it may be calculated by means of the following expression:

$$T_{ON}=\sum_{i=1}^{\infty} i \cdot q \cdot (1-q)^{i-1} \tag{9}$$

which serves to calculate the average cell transmission times (CTUs) that the source is at ON state. In each term of the addition, *i* is the CTUs the source remains at ON state, and $q(1-q)^{i-1}$ is the probability that the source remains *i* CTUs at ON state. Thus,

$$T_{ON}=...=\sum_{i=1}^{\infty} i \cdot q \cdot \underbrace{(1-q)}_{k}^{i-1}=\Sigma\, i\, q\, k^{i-1}=q \cdot \frac{d}{dk}\,[\Sigma\, k^i]=q \cdot \frac{d}{dk}\left(\frac{k}{1-k}\right)=\frac{q}{(1-k)^2}=\frac{1}{q} \tag{10}$$

In a similar way, we may obtain the average sojourn time at OFF state,

$$T_{OFF}=\sum_{i=1} i \cdot p \cdot (1-p)^{i-1}=\frac{1}{p} \tag{11}$$

If we now use the result obtained in [Bolla 96],

$$T_{ON} = B \qquad T_{OFF} = B \cdot (b-1) \tag{12}$$

Therefore, the time a given ID is busy, depends on the number of cells per PDU and the ratio of link capacity to PCR through parameter B (see equation 3). On the other hand, the time the source is not sending a PDU also depends on B and, in addition, on the burstiness (b) of the source.

Equations 3 and 12 have some implications on ID occupancy. As this parameter is related with the sojourn time at ON state of each source, we may deduce that for a given SCR, the higher the PCR, the less the source remains at ON state. In fact, maintaining the SCR while increasing the PCR implies the source is more bursty, and thus, sends all cells of the PDU in a smaller time. Furthermore, if the average PDU length decreases, so does the ON sojourn time, and as a consequence, the ID is freed faster and may be used by another PDU. Therefore, these results lead us to deduce that the best ID sharing might be obtained for bursty sources.

The discussion of the behavior of one source through the previous expressions may help in studying the behavior of the whole system, which is the topic of the following sections.

6.3.3 Simulation Environment

The scenario used for these simulations is represented in Figure 23.

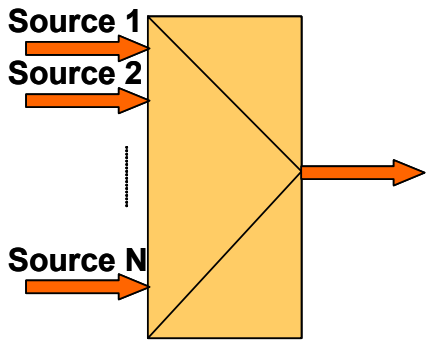

Figure 23. Simulated scenario for PDU ID dimensioning

In this scenario, each source just sends one PDU at any given time. In our simulator, the output queue is modeled as a counter of the number of simultaneous PDUs for each slot.

During all the simulations, the number of sources is varied depending on the average traffic introduced to the switch so as to assure that the losses only occur due to running out of identifiers and not due to surpassing link capacity. This assumption is possible if we consider that there is enough buffer space at the switch to absorb the burstiness due to the aggregation of sources. The simulated time is 10^{10} μs in all cases.

The reference source is characterized by the following parameters. At peak cell rate (PCR), the source transmits one cell out of fifteen, i.e. for an STM-1 link the PCR is 10 Mbps. The average traffic rate introduced by each source is 0.5 Mbps. The number of sources ranges from 100 to 300 to study the behavior of the mechanism for mid and high loads without reaching instability. PDUs are composed of an average of 5 cells and the sojourn time at ON state follows a geometric distribution. The OFF state also follows a geometric distribution with a mean of 1425 cells, which was chosen to obtain an average of 0.5Mbps per source. Any variation with respect to these parameters will be noted when presenting the results.

The following section presents the results obtained with this reference source and the comparison with other scenarios where one or more parameters in the source and the group are changed. In this way, we are able to study the dependence of the PLP on each of these parameters. The comparison between the simulated results and the theoretical ones are also presented in some representative cases. Part of these results were presented in [Mangues 00a] and [Mangues 00b].

6.3.4 Results

The distribution of the number of simultaneous PDUs for the source we take as reference is presented in Figure 24. Logarithmic scale has been chosen for both axis to provide a further detail for the range of values of interest, i.e. between 16 (4 bits) and 128 (7 bits) identifiers. We focus on these values because few bits in the ID provide low losses due to running out of ID (see Figure 25). Statistically non-significant values have been removed from the figure. That is, only points that present 95% confidence intervals smaller than one tenth of the value to represent are shown in the graph.

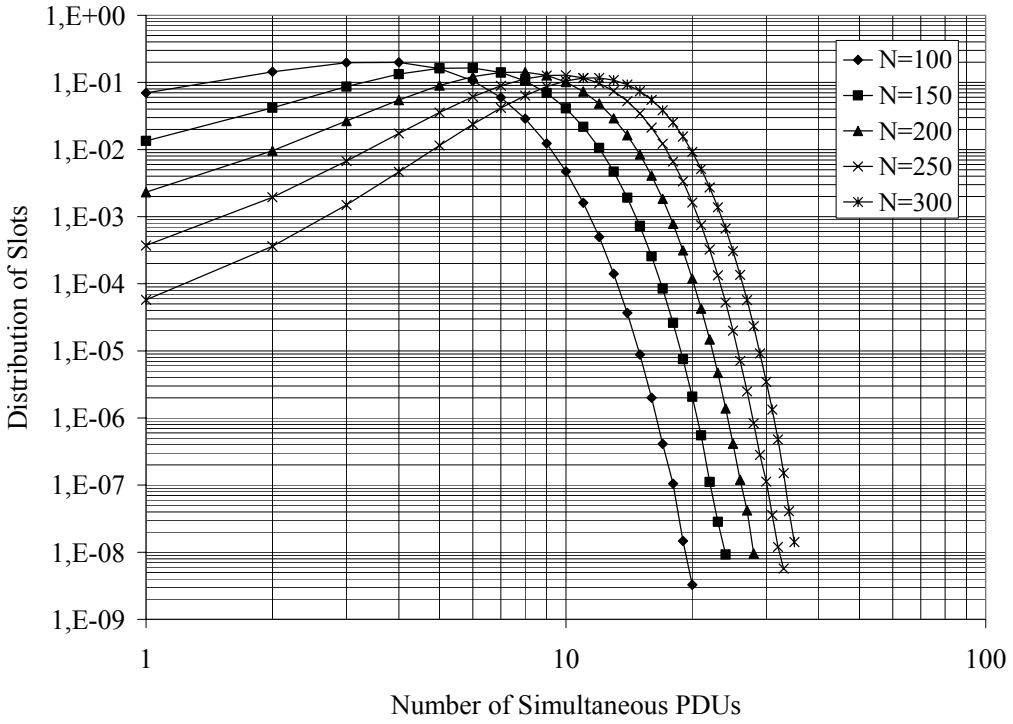

Figure 24. Distribution of the number of simultaneous PDUs at the output port of a switch where merging occurs. The reference scenario is average=0.5Mbps per source, 5 cells per PDU, and PCR=10Mbps. Each curve corresponds to a different number of sources (N).

Each curve corresponds to a different number of sources, and thus, different aggregated average loads at the switch, ranging from 50 Mbps for N=100 to 150Mbps for N=300. And the average number of simultaneous PDUs is 4.01 for N=100, 6.02 for N=150, 8.03 for N=200, 10.03 for N=250, and 12.03 for N=300. These results seem logical in the sense that, as there are more sources sending, the probability of having more PDUs simultaneously crossing the switch increases. Additionally, the peak of the curve moves to the right when N is increased. And, at the same time, the probability of having a low number of simultaneous PDUs decreases for the same reason. Furthermore, the range of possible values of simultaneous PDUs also increases with N.

As for the second step in the methodology, Figure 25 represents the probability that an arriving PDU does not find a free ID at the switch. Only values that present small 95% confidence intervals are represented. A comparison with the analytical expression commented above (equation 2) is also presented. The goal of the comparison is to study

the goodness of the fit between the results obtained through simulation and the theoretical expression for the simulated range of values. The PDU loss curves obtained by applying the analytical expression are labeled as *N (theor)* in Figure 25. The parameters appearing in this expression were mapped to parameters in our simulation to obtain such curves. A burst is taken to be a PDU in our simulation, *n* corresponds to the number of sources (N in the figures), *m* corresponds to the average number of simultaneous PDUs sent by the sources to the same output port, and *h* is the number of IDs. From the analysis of the expression it follows that each of the terms that are summed corresponds to the fraction of slots in which there are a given number of bursts being transmitted. In our case a burst corresponds to a PDU, therefore, Figure 24 represents each of the terms being summed. And the sum from *h* up to *n-1* is exactly how we obtain the graph of the PDU loss probability. It may be observed that the results of the simulation and those of the expression coincide for those values with small 95% confidence interval, which are the only values represented for curves obtained through simulation.

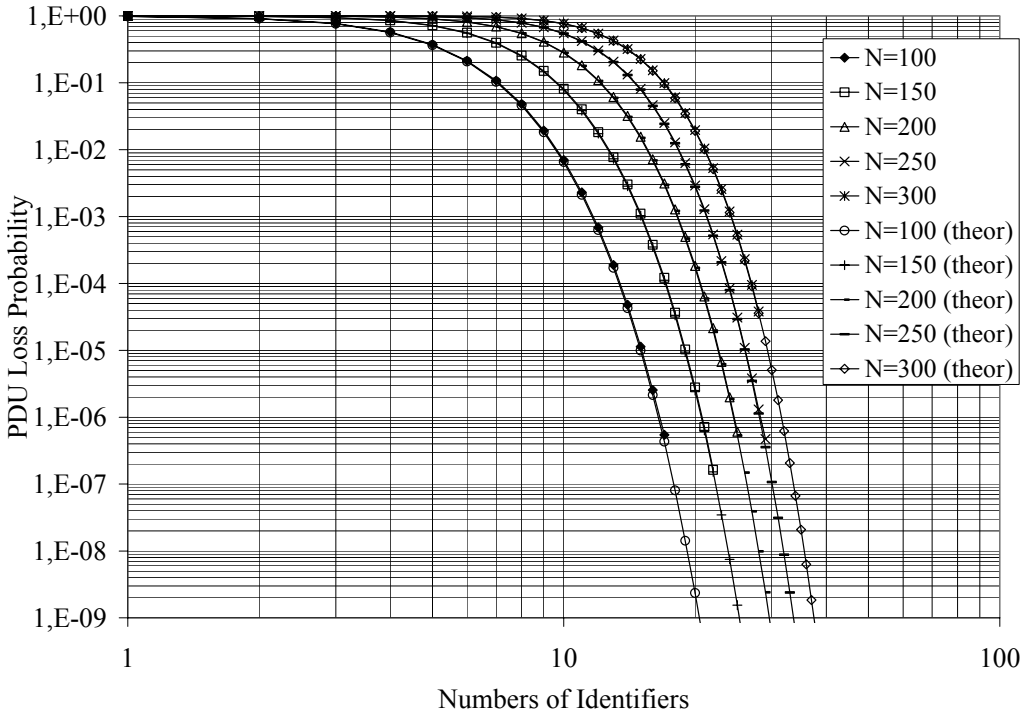

Figure 25. PDU Loss Probability due to running out of identifiers. Reference scenario is: average per source = 0.5Mbps, mean PDU length = 5 cells, and PCR = 10 Mbps.

It may also be observed that as N increases, so does the number of required identifiers to obtain the same PLP due to running out of identifiers. For instance, 17 IDs are required to obtain a PLP of 10^{-6} approx when N=100, whereas 32 are required to obtain the same PLP for N=300.

Another characteristic that may be observed in Figure 25 is the linearity (when using the logarithmic representation) of the PDU loss probability in the range of values of interest for the IDs. Therefore, the curve may be divided into three main linear regions. The first one starting at low ID values is flat, which tells that the number of IDs is not enough for such kind of traffic and group characteristics, as the PDU loss probability is 1. The second region, the one whose characterization is our main concern, would be a line going from the number of IDs where the curve starts to bend up to a number of IDs equal to N-1. The third region is characterized by a vertical line starting at nID=N, meaning that it is nonsense to use more IDs than sources in the group, because no losses occur due to running out of identifiers when nID≥N. The first two regions of the approximation are represented in Figure 26.

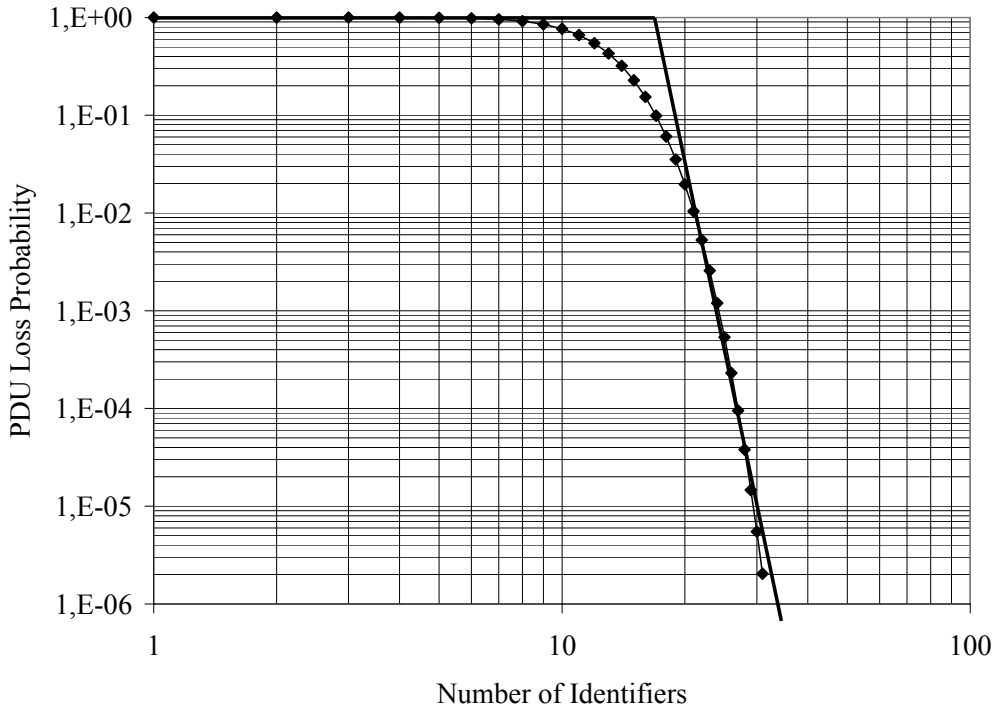

Figure 26. Linear approximation of the PDU Loss Probability (PLP) curve

Of course, such an approach is a rough approximation of the actual curve, but this characterization might allow a CVC switch to obtain, by means of a simple expression, the number of required IDs as a function of group and traffic characteristics and accepted PDU loss probability during connection establishment.

And finally, for the third step of the methodology, we proceed in a similar way, but this time by adding the rest of the values, i.e. from *0* up to *h-1*. We obtain the probability that an arriving PDU to the switch is correctly multiplexed and forwarded through the output port because it was able to allocate a free multiplexing ID. These results give us an idea of the throughput of PDUs.

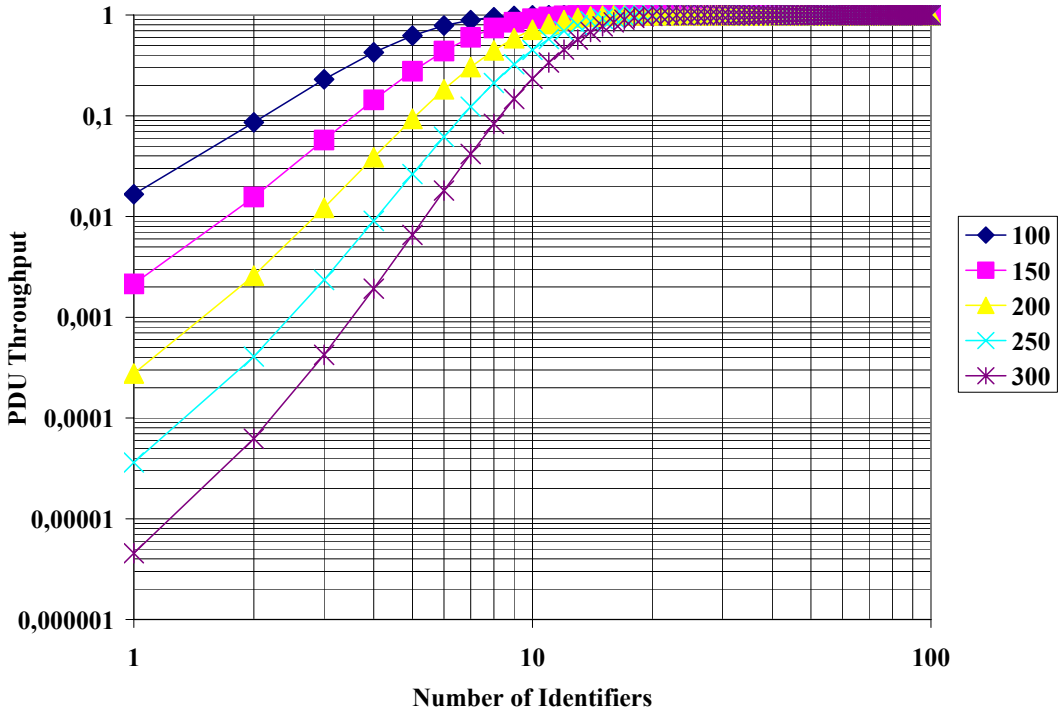

Figure 27. Throughput obtained in the reference scenario

Just remark that, for this scenario, with 16 identifiers, PDU throughputs near 90% are obtained for the highest aggregated load case (N=300). These curves also show that even in case of using very few identifiers, throughput values are far from 0, and thus, they justify the interest of ID sharing among PDUs. On the other hand, for the region of interest, i.e. that in which we obtain high throughputs with a few identifiers, more details

are provided by curves obtained in the second step, and thus, we mainly focus on those kinds of curves for ID dimensioning.

The main conclusion that may be drawn from these results is that with a few bits (between 4 and 6) and by using the PDU ID strategy, low PDU loss probabilities may be obtained even with a high number of sources. This scenario, which represents a possible scenario in future group communications, shows the advantages of PDU ID over Source ID multiplexing. These conclusions are further studied in the following sections.

6.3.4.1 Average cell rate

The first parameter under study is the average cell rate of each source. For this reason, simulations with different average cell rate per source are compared with the results obtained for the reference scenario. Notice that this parameter indirectly determines the number of sources being simulated. As a matter of fact, our study focuses on situations with high loads, where IDs need to be shared more efficiently. For this reason, the number of sources chosen depends on the average of each source and the capacity of the link. Recall that, for simplicity, in our simulations we suppose that there is only one group that is using all the capacity of the link. The simulated aggregated load ranges from 33% to 100% of the output link capacity on the average. This is the reason why the number of sources (N) varies from 100 to 300 when the average per source is 0.5 Mbps (reference scenario).

PDU loss probability results were obtained for a number of sources ranging from 500 to 1500 with an average traffic per source of 0.1 Mbps. This new average value is obtained by varying the mean sojourn time at OFF state. The rest of the traffic parameters are the same as those in the reference scenario. These curves are represented in Figure 28.

The same observations stated above for the reference scenario apply for this new scenario. However, notice that there are far more sources than before, but the number of identifiers is the same for the same aggregated load. Therefore, the advantages of ID sharing are more evident in this case.

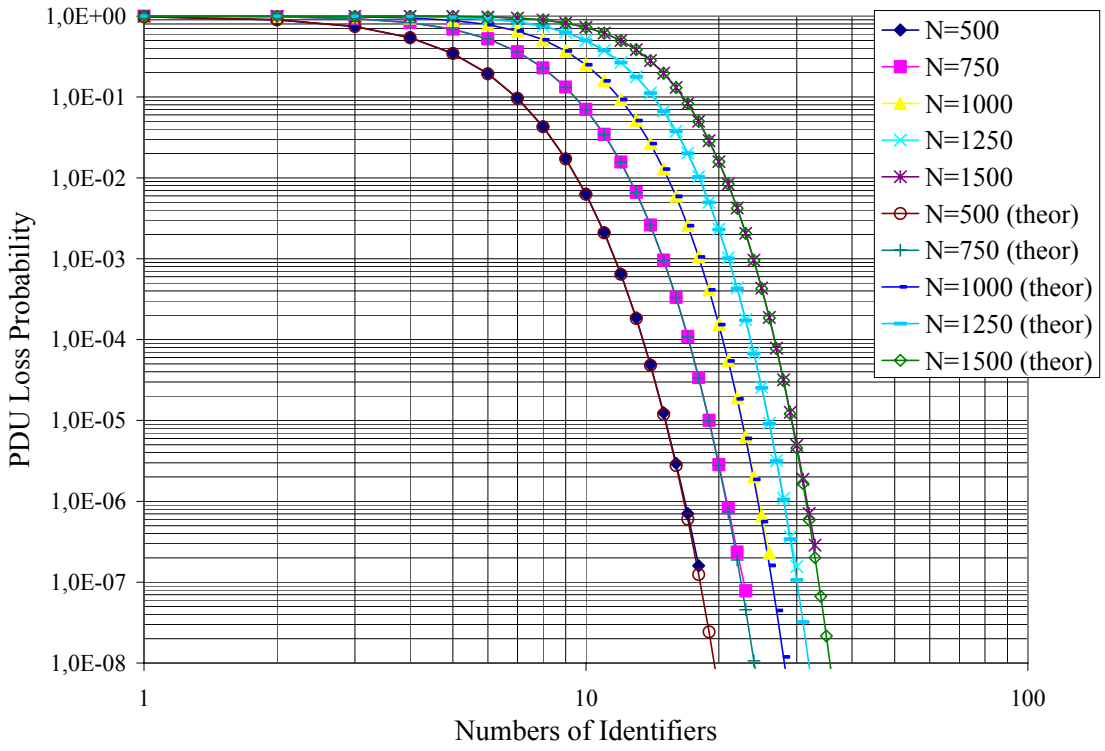

Figure 28. PLP curves for average per source = 0.1 Mbps (rest of parameters are the same as in the reference scenario)

The simulation results are also compared with the analytical ones (Turner's expression). Both curves overlap for statistically significant values.

However, simulations with high average per source (e.g. 5 Mbps) showed a difference between the analytical and the simulated curves (see Figure 29). It may be observed that, the more the number of sources, the more the simulated and analytical curves resemble.

The reason of this difference may be found in that statistical expressions are based on averages of high number of occurrences of an event. Nevertheless, with 5 Mbps, there are few sources, and thus, there is a small number of simultaneous PDUs traversing the switch. As a consequence, the simulated scenario and the analytical expression do not lead to exactly the same results, unlike in the previous cases, where there is a high number of sources, and thus, of simultaneous PDUs.

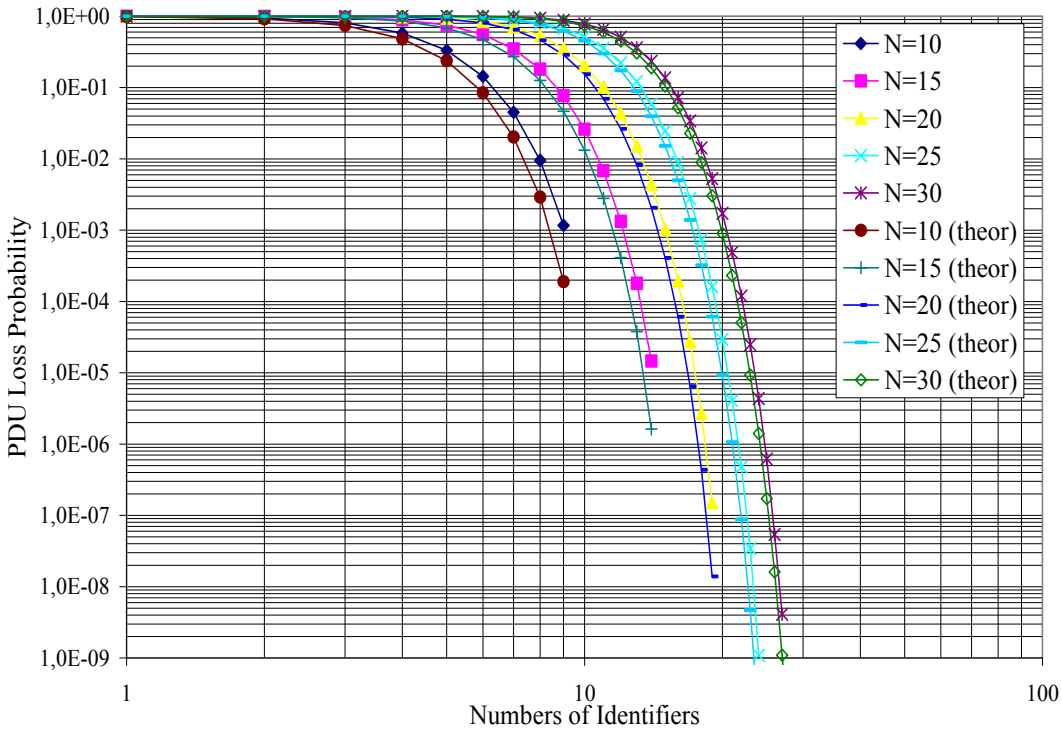

Figure 29. PLP curve for average per source = 5 Mbps

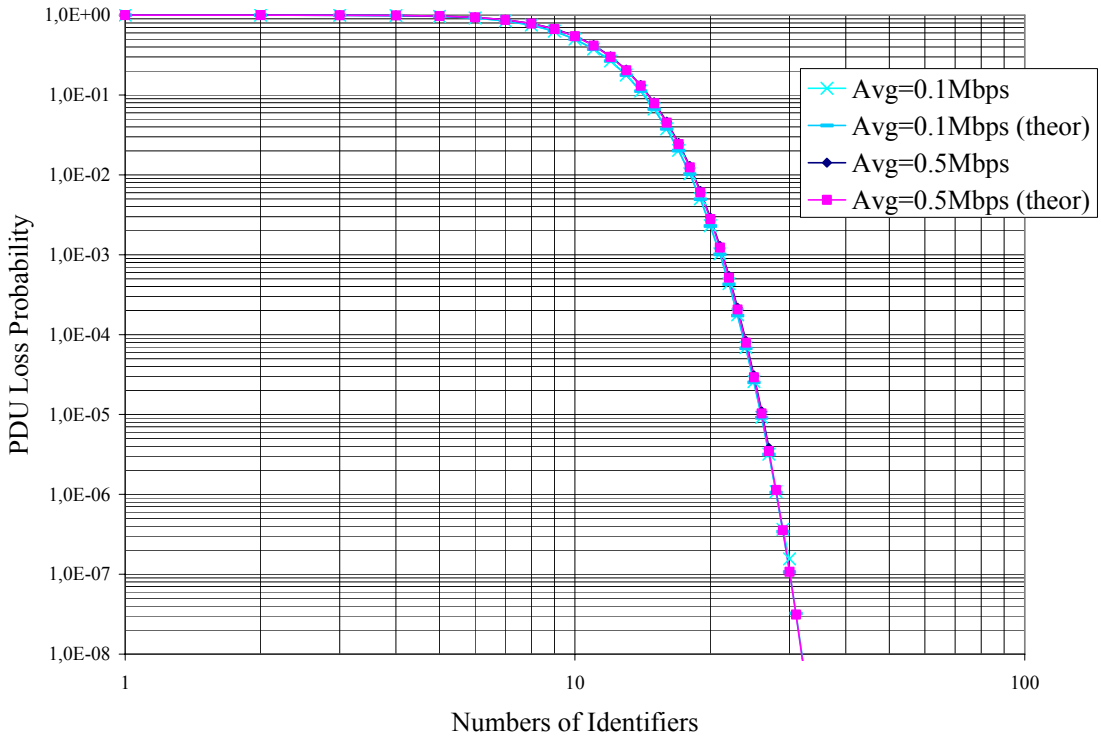

Figure 30. Comparison of PLP curves for averages per source = 0.1 Mbps and 0.5 Mbps

Finally, the comparison between the results presented in Figure 25 (reference scenario) and those obtained for the scenario with average per source=0.1 Mbps showed an almost imperceptible variation in the curves when the aggregated average load is the same (see Figure 30). In particular, this figure presents four curves, two are obtained through simulation (Avg=0.1Mbps and Avg=0.5 Mbps) and the other two by means of the analytical expression (Avg=0.1Mbps(theor) and Avg=0.5 Mbps(theor)). They all correspond to an aggregated load of the 83%, which corresponds to N=250 for 0.5Mbps per source and N=1250 for 0.1Mbps per source. This suggests that the aggregated average load is a fundamental parameter for the calculation of the number of identifiers. This result is further confirmed in section 6.3.5.

6.3.4.2 Cells per PDU

From the analytical expressions of the average sojourn times at ON and OFF states (equation 12), it may be deduced that there is a clear influence of the number of cells per PDU on the time IDs are busy. Recall from equation 3 that B depends on the PDU length (or number of cells per PDU). Therefore, this should have an influence in the PLP values obtained. However, the results obtained for reasonable values of PDU sizes are in contrast with the previous statement.

Figure 31 presents a comparison of the curves obtained for N=300 (with the rest of parameters as in the reference scenario) when varying the mean PDU length. The simulated values are 5, 10, and 15 cells per PDU, which correspond to reasonable mean values according to current Internet traffic. In fact, most IP packets would fit in less than 5 cells because they are due to TCP acknowledgements, as most communications nowadays correspond to client-server interactions, or for peer to peer applications, they correspond to content provider to content consumer interactions.

As it may be expected, the longer the PDU, the longer the IDs are occupied, and the more IDs are required. However, the variation between these curves is not very high. Thus, it may be observed that the curves show the same behavior (they all have the same shape). The only difference is a slight drift. Therefore, in the rough linear model we proposed to describe the behavior of these curves, it seems that the dependence of the equation of the line in the region of interest would be on the position of the point where

the curve bends and not on the slope. Anyway, for reasonable mean values, such dependence would not be very strong.

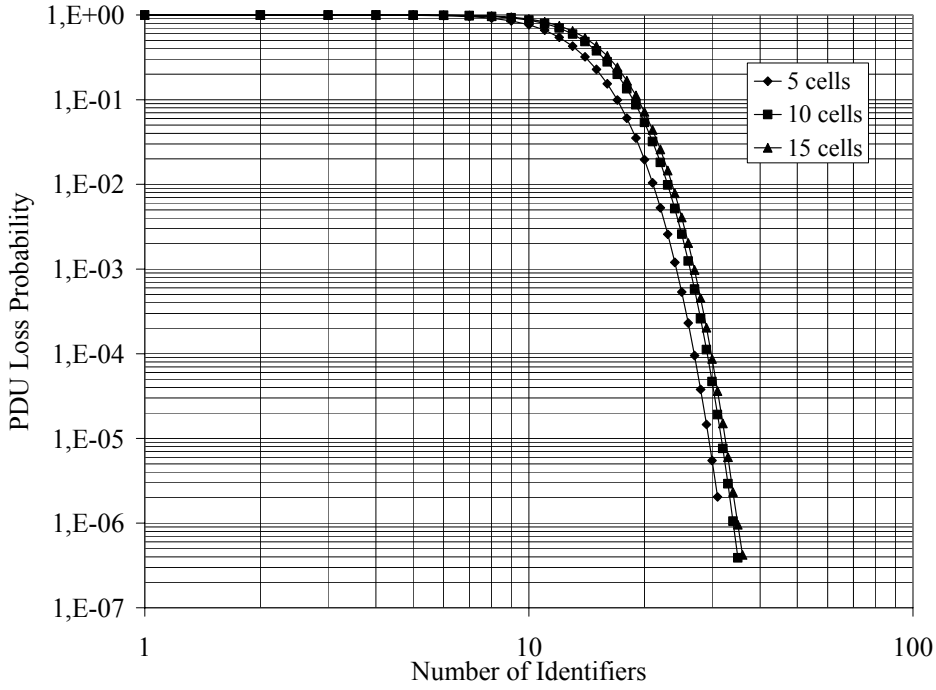

Figure 31. PDU Loss Probability comparison varying the mean PDU length. Average per source=0.5 Mbps, PCR=10 Mbps, and N=300.

6.3.4.3 Peak Cell Rate (PCR)

The figures in this section are devoted to study the dependence of the PLP curves on the PCR. In fact, in these results, the parameter that is actually compared is the burstiness of the sources, and not directly the PCR. This is due to the relationship between the PCR and the burstiness of the traffic introduced by the source. That is, if we maintain the same average traffic per source and rest of parameters and we vary the PCR, the same number of cells per PDU is sent, but at a higher speed. Therefore, the PDU lasts less and as a consequence it is using an ID during less time, making it possible for other sources to get that ID. Recall that throughout this thesis, we understand the burstiness of the source as the ratio PCR to SCR. In particular, the curve labeled as 2Mbps (i.e. PCR=2Mbps) corresponds to a burstiness of 4, 10 for 5Mbps, 20 for 10Mbps, 60 for 30Mbps, and 300 for 150Mbps, because for all the simulated curves the average rate is 0.5Mbps. Figure 32

shows the results obtained when varying the PCR. Recall that for the reference scenario PCR=10Mbps. Notice also that all the curves correspond to N=250 sources.

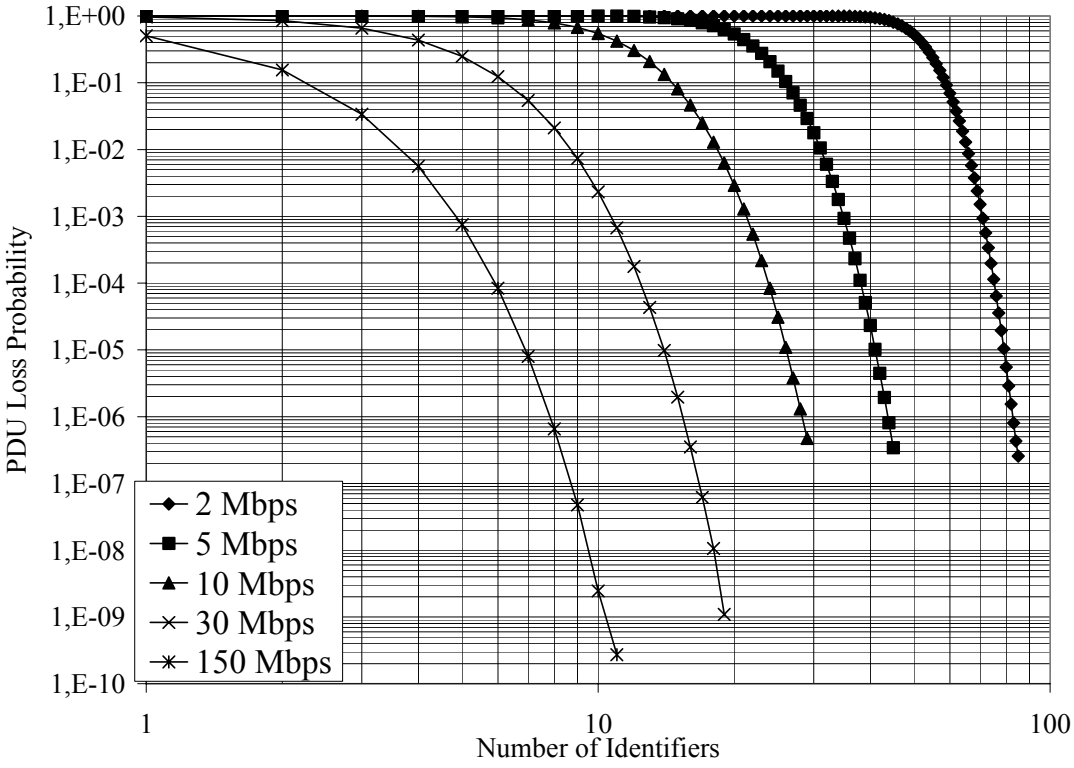

Figure 32. Comparison of PLP curves for various PCR values when N=250 (rest of parameters are the same as in the reference scenario)

The results show a high dependence of the PLP curves on the PCR (or burstiness) of the sources. In our linear approximation, both the slope and specially the y-intercept of the line in the region of interest are modified. For ID dimensioning purposes, the most important variation is in the intercept, because slope change is not very significant.

Other simulations have been done with different average per source, but with the same burstiness and average aggregated load. These results are presented in Figure 33.

It may be deduced from Figure 33 and the preceding ones in this section, that the number of IDs required does not exclusively depend on the burstiness, because in that case, the two curves in Figure 33 would overlap. It depends on the PCR and with less importance on the PDU length. That is, given a PCR and an average PDU length, if the

average aggregated load is the same, the number of identifiers required are the same. This is exactly what could be observed in Figure 30. Recall from section 6.3.2, that PCR and PDU length (jointly with the link capacity) determine the ON sojourn time of a given source. Therefore, our results confirm what was expected from the theoretical calculations.

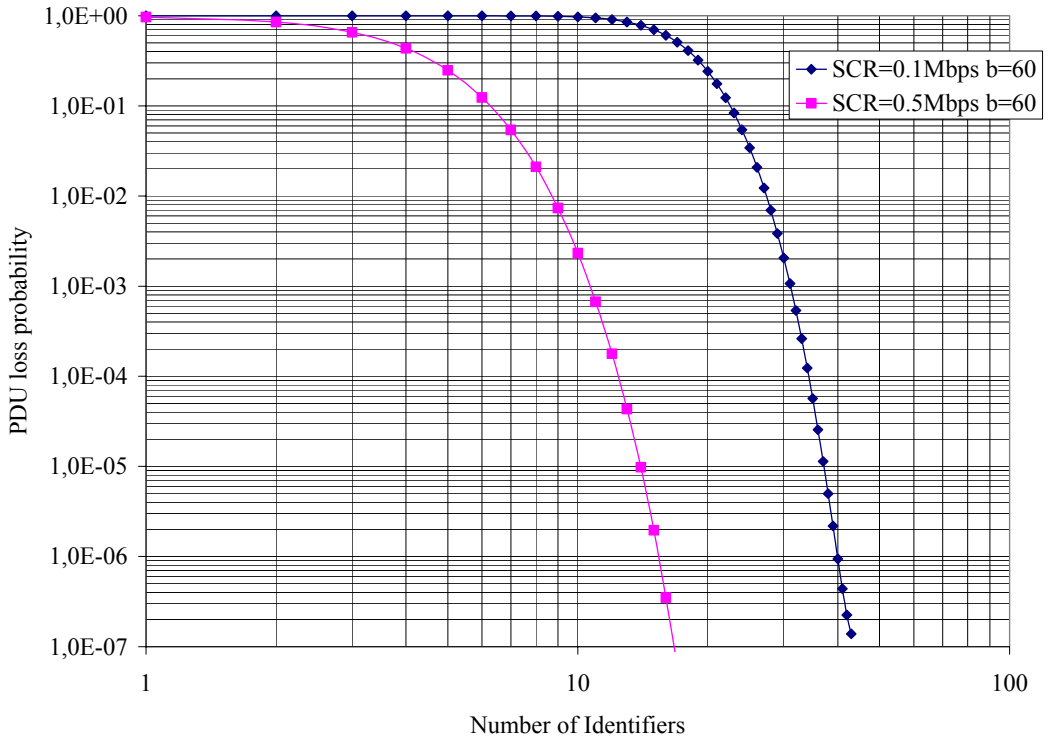

Figure 33. Comparison of PLP curves for various SCR=0.5Mbps and SCR=0.1Mbps with same burstiness=60 and aggregated load=125 Mbps

6.3.4.4 Heterogeneous parameter comparison

Finally, Figure 34 presents a comparison of the reference scenario described before with others in which one or two parameters are changed with respect to the reference one. These results were obtained for N=250 sources, which produce an aggregated traffic of 125Mbps, as the average traffic per source is 0.5Mbps.

We first focus on the curves that just vary the PCR while maintaining the rest of the reference parameters. They are labeled as *PCR=2Mbps, avg=0.5* (which corresponds to PCR=10 Mbps), *PCR=30Mbps*, and *PCR=150Mbps*. In this case, the range of possible

values is wider than in the PDU case. Recall from the above discussion that the curve *PCR=150 Mbps* corresponds to the most bursty sources, and *PCR=2 Mbps* corresponds to the smoothest simulated traffic, as all the rest of parameters are the same for all these curves.

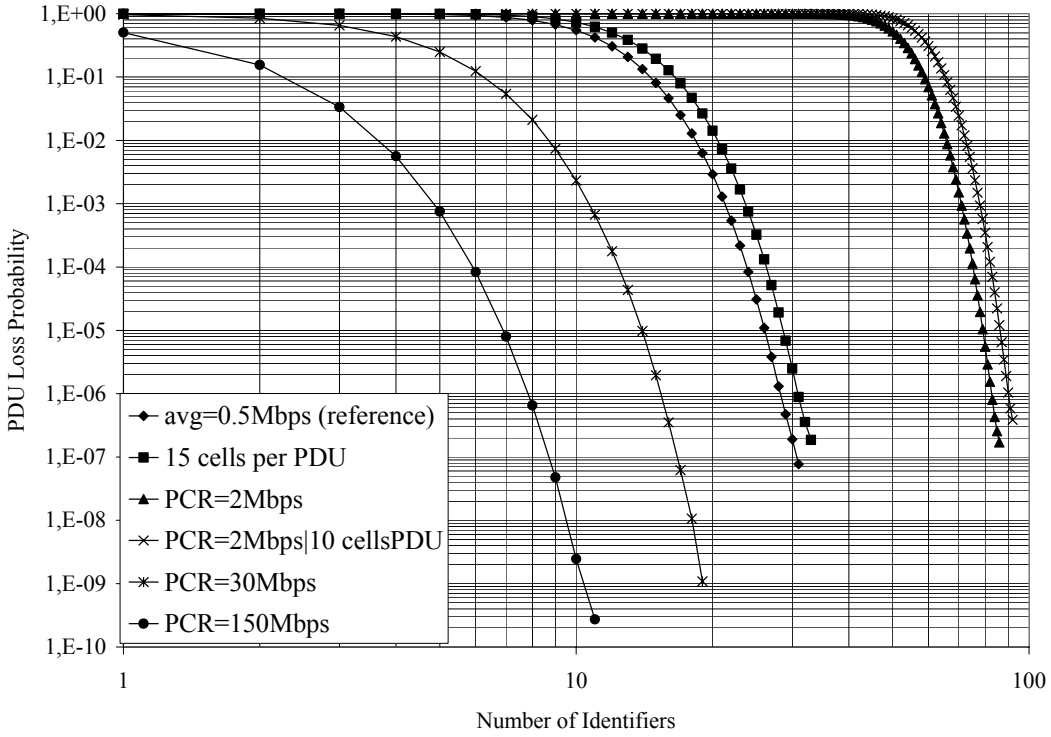

Figure 34. PDU Loss Probability comparison for different parameters

For instance, to obtain a PLP of 1e-6, the number of bits required for the PDU ID is respectively 7 (128 IDs), 5 (32 IDs), 4 (16 IDs), and 3 (8 IDs) for the 2, 10, 30, and 150 Mbps cases.

The same observations as in the PCR comparison apply in this case, that is, these results show a strong dependence of the drift of the curves with respect to the PCR. However, the slope of the line in the region of interest seems to show only a slight dependence on the PCR.

The curve labeled as *PCR=2Mbps|10 cells per PDU* serves to confirm that the PLP curve is more sensitive to PCR variations, i.e. to the burstiness of the traffic, than to the length of the PDUs. Its values are more similar to those of the curved labeled as *PCR=2Mbps* than to those obtained for the *15 cells per PDU* case.

We calculated the burstiness of the sources for which their PLP curves are represented in Figure 34. The burstiness (b) is 4 (=2Mbps/0.5Mbps) for the sources whose curves are labeled as *PCR=2Mbps*, and *PCR=2Mbps|10cellsPDU*, 20 for *avg=0.5* and *15 cellsPDU*, 60 for *PCR=30Mbps*, and 300 for *PCR=150Mbps*. For these new curves, and to obtain a PLP of 1e-6, the number of bits required is respectively, 7, 5, 4, and 3, which are the same as those obtained when the focus was on comparing the PCR values. Therefore, this observation confirms that, with these traffic parameters, the influence of the PCR is much bigger than that of the PDU length.

The diversity in scenarios and requirements for different groups also shows the advantages of having flexible ID size negotiation, such as the one offered by CVC. For instance, in multimedia group communications more losses could be accepted for video than for audio, and different number of sources would require different number of IDs.

6.3.5 Erlang-B approach

This latter approach tries to provide a more analytical way of dimensioning the CVC communication, and as such, it is extendable to a wider range of scenarios. Turner's expression provides very similar results to those obtained through simulation but its main problem is that its parameters are not directly related to those of the sources. As a consequence, dimensioning at connection establishment by means of this expression is difficult. On the other hand, the parameters used in Erlang calculations in our test conditions are easily derived from those characterizing the sources. Therefore, dimensioning is simpler than in the previous case. Furthermore, Erlang expressions provide results that are very similar to those obtained with Turner's expression and through simulation for the conditions under test. Therefore, this section describes an analytical approach to dimensioning, that, as a consequence, applies to a wider spectrum of cases than simulated results.

6.3.5.1 Motivation

There are some characteristics of our system that lead us to compare it with a loss system and thus, to use Erlang-B expressions to describe its behavior. But in our case, the goal is not to calculate the number of circuits given some input traffic characteristics, but to

dimension the multiplexing identifiers. The operation of both systems is the same, as PDUs are discarded if they do not find a free resource (ID) in the same way as calls are rejected in case all circuits are already allocated.

The rest of characteristics of the system match as follows with those of the original loss system. Our sources generate PDUs whose inter-arrival times follow an exponential distribution (or a more appropriate term in the discrete domain, geometric). Furthermore, as we are trying to stress the CVC mechanism to evaluate its scalability, a high number of sources is tested in most cases, which is usually much bigger than the number of resources to share (mux IDs). Therefore, this scenario resembles the theoretical one of a Poisson arrival process, which is characterized by having an infinite number of sources. Additionally, a call is matched with a PDU in our model, and the exponential distribution of its duration corresponds to the geometric distribution of the PDU length.

In conclusion, the original circuit dimensioning problem in a link is now translated into the PDU ID dimensioning problem of a CVC connection.

6.3.5.2 Discussion of the Erlang-B expression

From the above discussion, it may be observed that the system under study is an M/M/c/c, i.e. PDU arrivals occur according to a Poisson process, the server/resource is busy during time intervals that follow an exponential distribution, there are c servers, and no more than c PDUs are allowed in the system. The parameters that characterize such a system are represented in Figure 35, where λ is the input rate, μ is the service rate of each server, and c is the number of servers/resources to dimension.

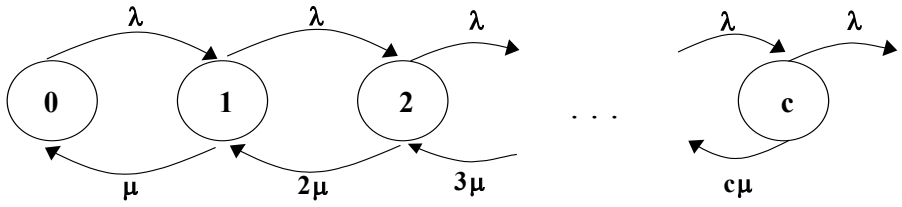

Figure 35. M/M/c/c loss system that represents the ID dimensioning problem

The calculation of the probability of being in a given state *n* leads to the following expression.

$$P_n = \frac{\frac{A^n}{n!}}{\sum_{i=0}^{c} \frac{A^i}{i!}} \quad (13)$$

If we are interested in the loss probability due to running out of resources, the focus is on the last state of the Markov chain. The reason is that if the system is in this state, an arriving PDU does not find free resources and is lost. Therefore, the loss probability under calculation directly corresponds to the probability of being in state *c*.

Notice that it just depends on c (the number of resources to dimension), and A (the traffic intensity introduced by all the sources in the system). This latter parameter is calculated as the ratio among the input rate (λ) and the service rate (μ). A discussion of the implications of such characteristic in our system is given in the following section.

6.3.5.3 Translation of source parameters into loss system parameters

The parameters that characterize the sources are the SCR, the PCR, and the PDU length, i.e. the number of cells per PDU. These are available to each source at connection establishment. Other parameters that must be taken into account to fully characterize the loss system are the number of sources and the capacity of the link. The translation of the parameters into the Erlang model requires the calculation of the traffic intensity introduced to the system by all the sources, which is measured in Erlangs. In the original circuit dimensioning system the traffic intensity introduced to the system is obtained as follows:

$$A\ (\text{Erlangs}) = \frac{\lambda}{\mu} = \frac{\text{Number of calls}}{\text{second}} \cdot \frac{\text{seconds}}{\text{each call}} \quad (14)$$

In our multiplexing ID dimensioning system, and according to the mapping of parameters above, the number of calls per second is translated into the number of PDUs per second, and the length of the call is translated into the length of the PDU (expressed in seconds). Therefore,

$$\lambda = \frac{\text{SCR(Mbps)} \cdot 10^6}{424 \cdot \text{PDU\_length (cells)}}\ (\text{PDUs / second}) \quad (15)$$

As for the service rate, Figure 36 may help in explaining the expression presented below.

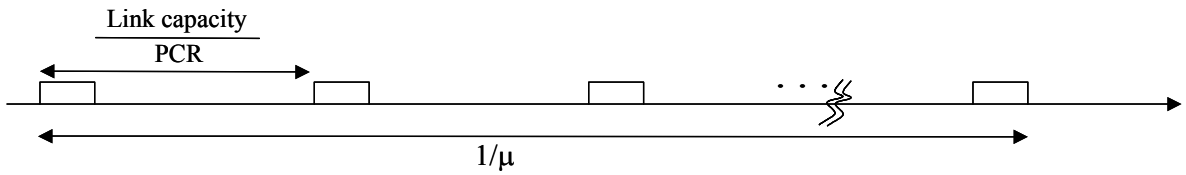

Figure 36. Parameters that characterize the transmission of a PDU

$$\frac{1}{\mu} = \left((\text{PDU\_length} - 1) \cdot \frac{\text{Link\_capacity}}{\text{PCR}} + 1 \right) \cdot \frac{424}{\text{Link\_capacity}} \text{ (seconds / PDU)} \qquad \textbf{(16)}$$

and thus, the traffic intensity introduced by one source is

$$A_{1\ \text{source}} = \frac{1}{\text{burstiness}} \cdot \left(1 - \frac{1}{\text{PDU\_length (cells)}} \right) + \frac{\text{SCR(Mbps)} \cdot 10^6}{\text{PDU\_length (cells)} \cdot \text{Link\_capacity (bps)}} \text{ (Erlangs)} \qquad \textbf{(17)}$$

As a consequence, the total input traffic intensity to the system is

$$A_{total} = \text{Number\_sources} \cdot A_{1\ \text{source}} \qquad \textbf{(18)}$$

From the above expressions, the dependence on A translates in our study in a dependence on the burstiness (i.e. the ratio PCR/SCR) and, for the values of our interest, in a smaller dependence on the average PDU length. Notice that for the values of the parameters we chose in the simulations, the first term is, in most cases, much bigger than the second term of the sum in the $A_{1source}$ expression. This observation in addition to the results obtained through simulation lead us to confirm the conclusions that were already drawn from the results in the previous section, i.e. a strong dependence of the PLP on the burstiness (and implicitly on the PCR) and a smaller dependence on the PDU length for the values tested in our study.

Furthermore, in the notation used in this section, the average aggregated load corresponds to the total traffic intensity introduced to the system (i.e. A_{total}). By taking a look at equation 13, it may be concluded that in homogeneous scenarios, if parameters are found for different kinds of sources so that equation 18 results in the same value, the PLP curve behaves in the same way for all these different scenarios. And it does that

disregarding the particular traffic characteristics of each source. Therefore, the dimensioning process in all these scenarios leads to the same result.

The following paragraphs are devoted to present a comparison of the results obtained with Turner's expression, simulation and the Erlang-B approach in some selected cases. Other cases that were compared offered similar results.

6.3.5.4 Number of sources

The first comparisons were carried out to confirm that the three curves behave in the same way for different number of sources in the range of nID values that provide acceptable PLP values. Figure 37 and Figure 38 present the comparison for the reference scenario with N=150 sources and N=300 sources respectively. Recall that the reference parameters are: average load per source = 0.5 Mbps, Cells per PDU = 5 cells, PCR = 10 Mbps.

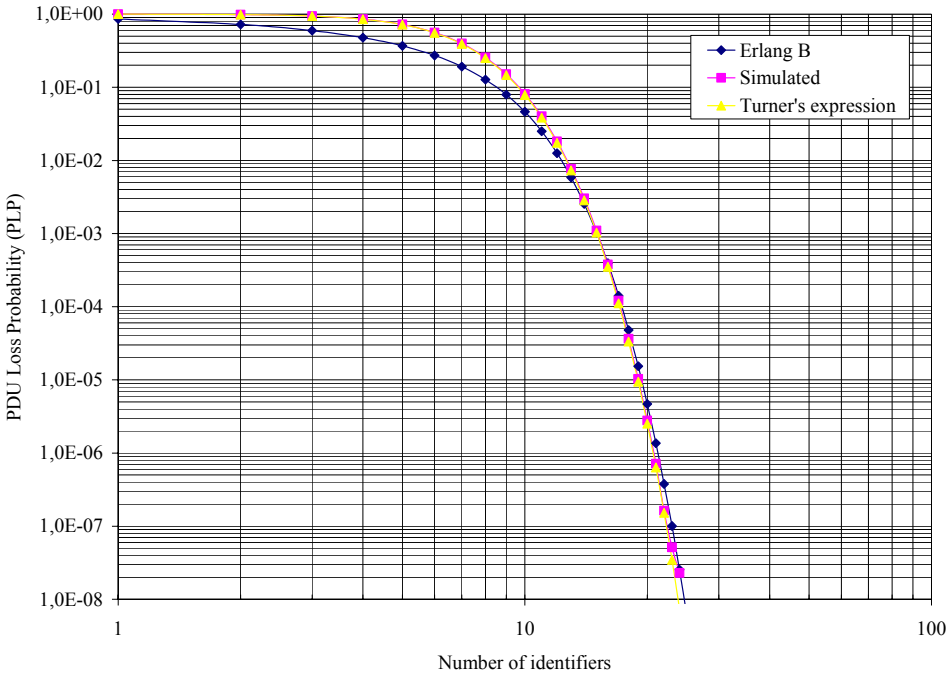

Figure 37. Comparison of PLP curves obtained in three different ways. Parameters of the scenario: Reference scenario, N=150 sources

It may be observed that there is a slight difference in the behavior of Erlang-B curve with respect to the other two. But this difference is concentrated in the region where the curves bend, where the Erlang-B expression provides a more optimistic value of the PLP

than that provided by the other two. On the other hand, the three curves overlap in the region where they show a steady decrease. Interestingly, this is the interval of nID values we are interested in when dimensioning the connection, because acceptable PLP values are obtained for the scenarios we focus on.

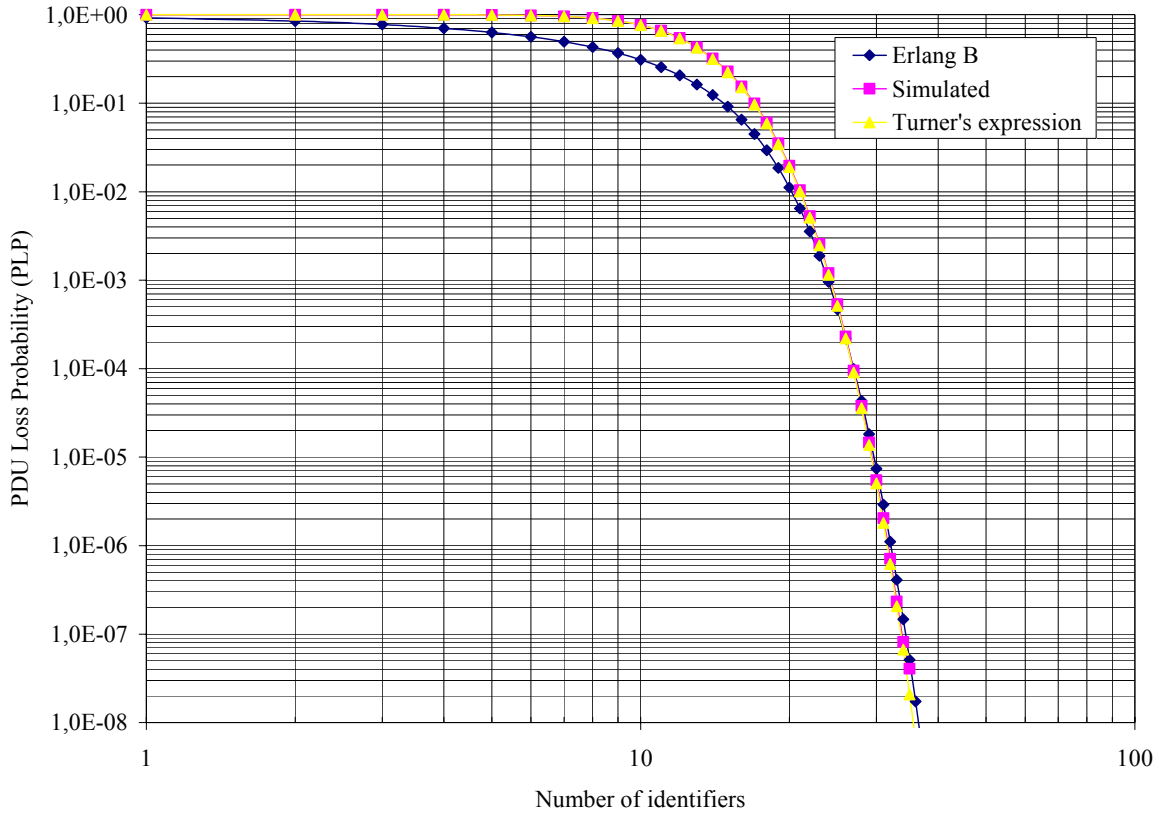

Figure 38. Comparison of PLP curves obtained in three different ways. Parameters of the scenario: Reference scenario, N=300 sources

Notice that in Figure 37, the number of sources is just 10 times bigger than the resources (multiplexing IDs) to dimension in the region of interest. This is far from approaching the infinite source Erlang-B theoretical scenario. In spite of that, there is just an almost imperceptible drift to the right of the Erlang-B curve.

In the following sections, there are other parameter values that characterize different scenarios with varying number of sources that also confirm the observations stated in the previous paragraphs.

6.3.5.5 Average cell rate

In this section, the sustainable cell rate (SCR) of the sources is varied with respect to the that of reference scenario. In this case, SCR=0.1 Mbps and N=750 sources (Figure 39) and N=1500 sources (Figure 40). The rest of the parameters are the same as in the reference scenario.

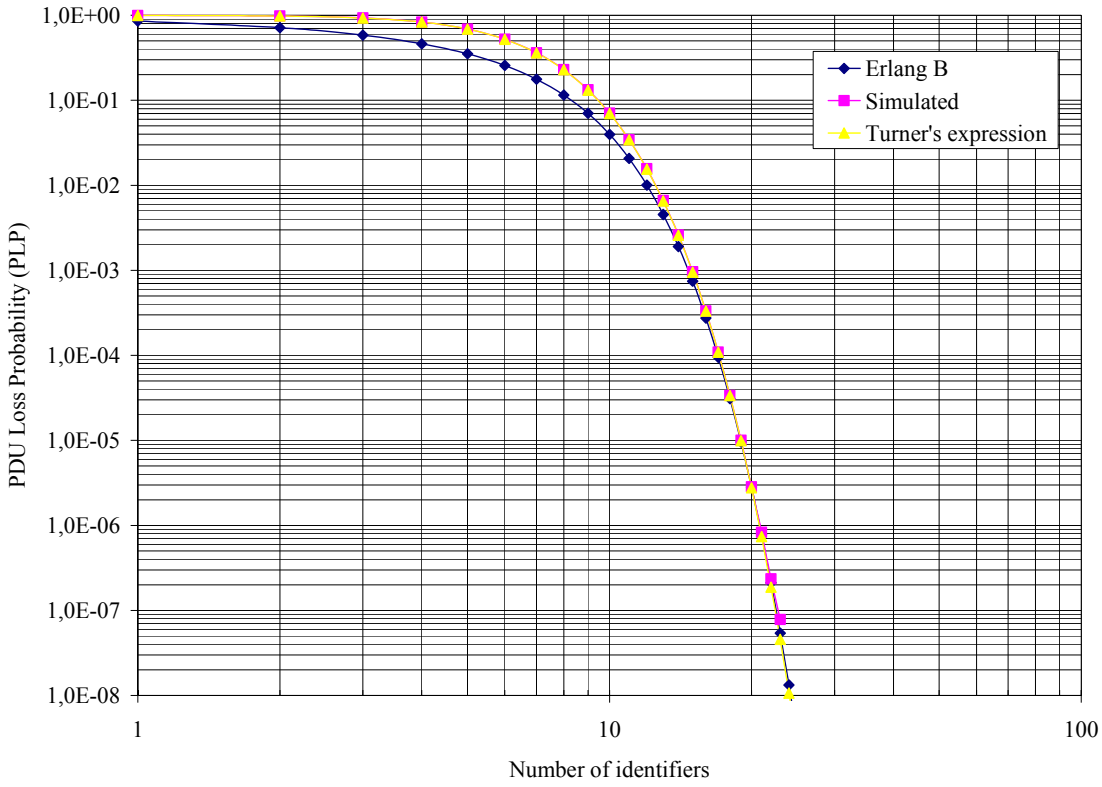

Figure 39. Comparison of PLP curves obtained in three different ways. Parameters: Avg per source = 0.1 Mbps, N=750 sources, rest of parameters same as in reference scenario

The same observations as in the previous section apply here. The three curves overlap in the region of interest. Therefore, this observation might allow us to conclude that the approximation made when matching our dimensioning problem with the circuit dimensioning one works for our purposes.

Notice that in this case, the number of sources is much bigger than in the previous one, and thus, this scenario is more similar to that with theoretical Poisson arrivals with an infinite number of sources that characterizes the M/M/c/c loss system whose behavior is

described by the Erlang-B expression. Notice also that, as the approximation is better, the curve almost perfectly overlaps with the other two, in the region of interest.

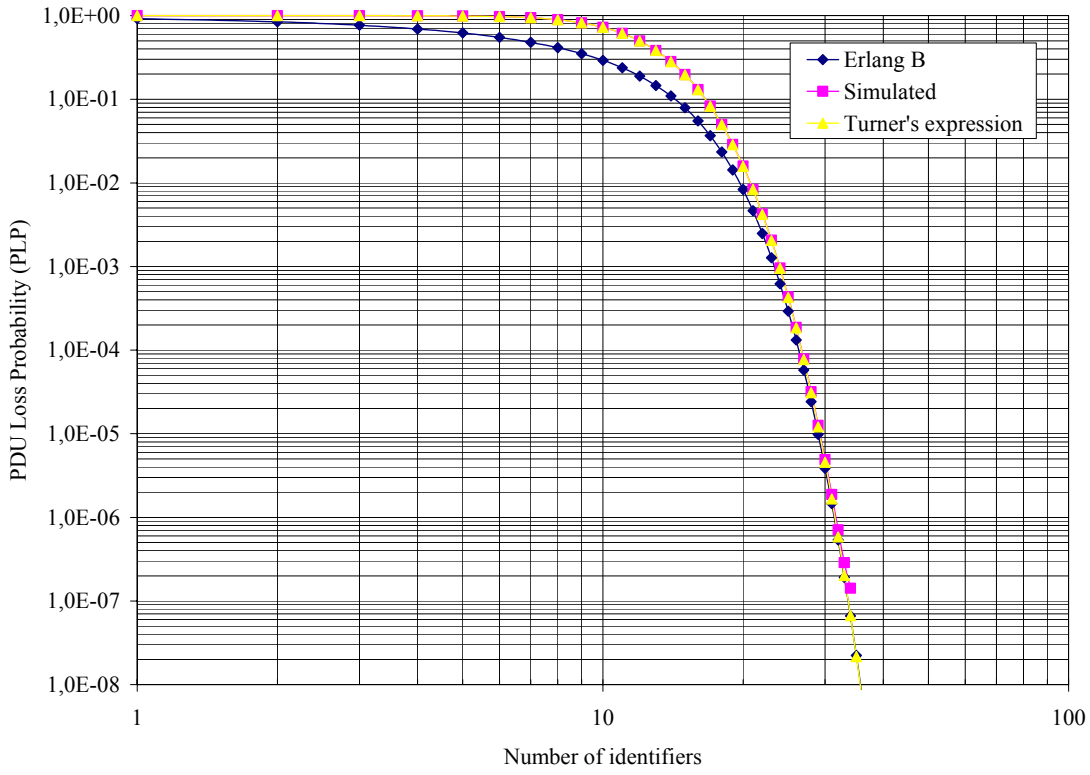

Figure 40. Comparison of PLP curves obtained in three different ways. Parameters: Avg per source = 0.1 Mbps, N=1500 sources, rest of parameters same as in reference scenario

Other comparisons were carried out with a number of sources in the order of the number of required identifiers. In this case, the approximation of the theoretical model does not hold. However, the Erlang-B curve follows the same trend as the other curves, but the resemblance is not as good as in the previous scenarios. A more pessimistic result is obtained with the Erlang-B expression in the region of values of interest. Nevertheless, in most cases, the difference between both curves would not be significant from the dimensioning point of view due to the round up of the number of identifiers to a power of two imposed by CVC.

6.3.5.6 Cells per PDU

Figure 41 also shows that the Erlang-B curve also follows the changes in the behavior due to a new average PDU length value, in this case 15 cells.

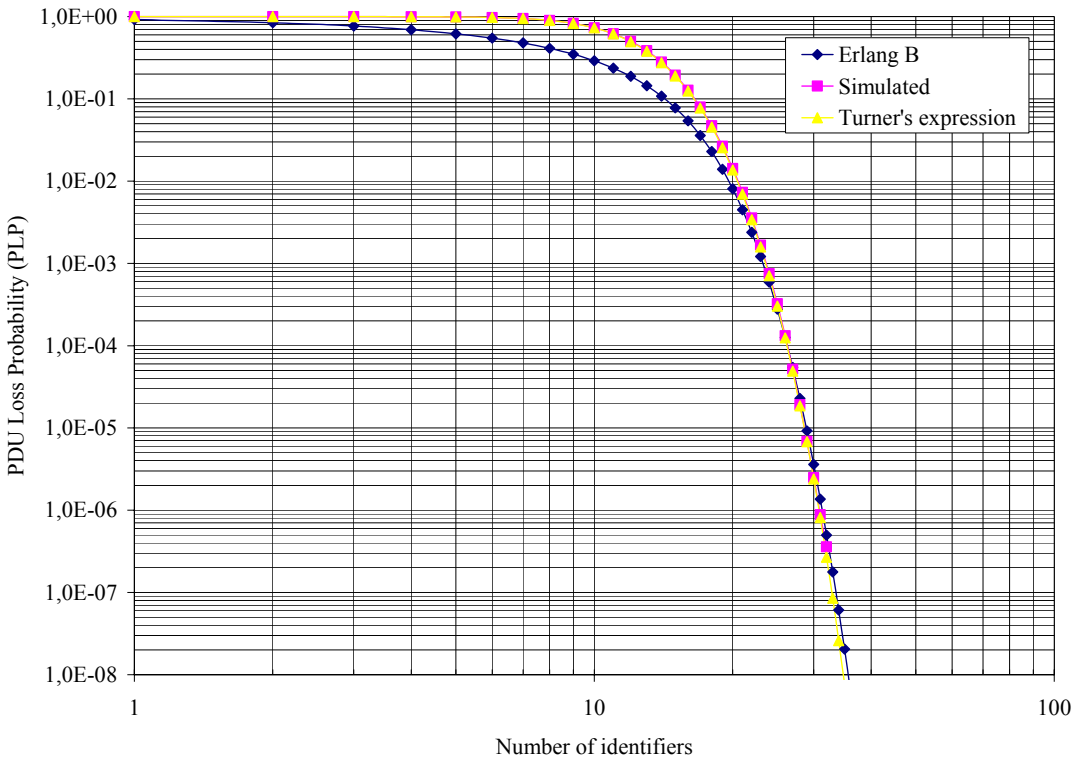

Figure 41. Comparison of PLP curves obtained in three different ways. Parameters: Cells per PDU = 15 cells, N=250 sources, rest of parameters same as in reference scenario

6.3.5.7 Peak cell rate (PCR)

The PCR comparison also leads to the same observations though the resemblance of the Erlang-B curve with the other two is not as good as in the previous comparisons for the range of values under consideration. Notice, however, that the range of values under test is very wide. Values near both ends of all the possible values are tested, PCR=2Mbps in Figure 42 and PCR=150Mbps in Figure 43. The latter corresponds to a case in which the cells of the PDU are sent back-to-back whereas in the former the PDU lasts more and thus more IDs are required on the average.

In both cases, the Erlang-B curve gives a slightly more pessimistic result in the region of interest. However, the reader should notice that for the dimensioning of the ID, just values that are a power of two may be chosen. Therefore, even in cases where the theoretical and the simulated curves are not exactly the same, the roundup to the closest power of two value leads to the same result of the dimensioning process.

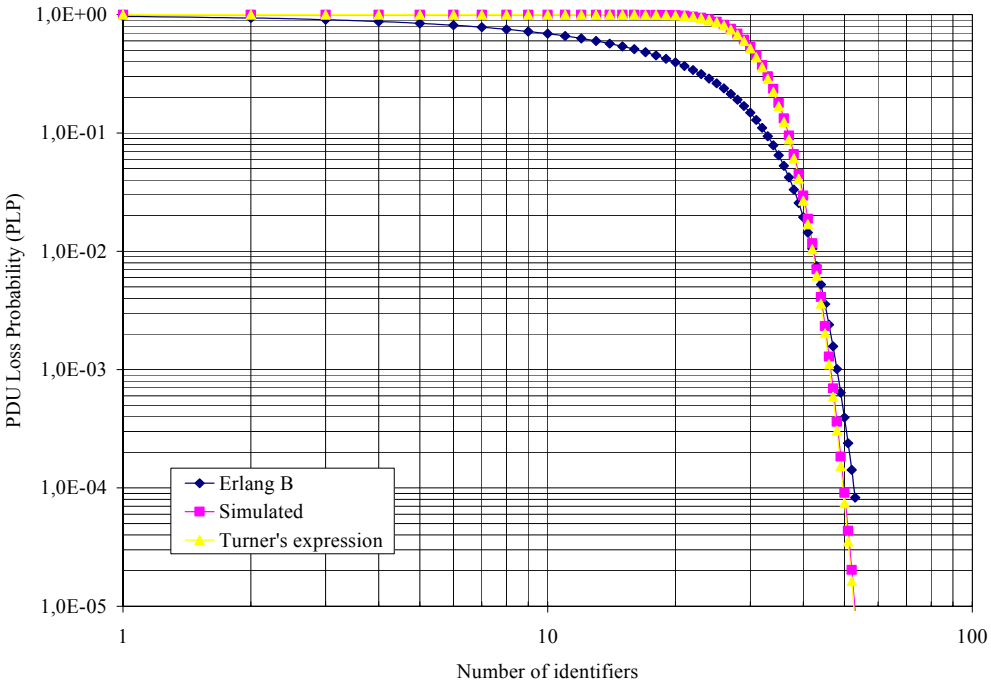

Figure 42. Comparison of PLP curves obtained in three different ways. Parameters: PCR=2Mbps, N=150 sources, rest of parameters same as in reference scenario

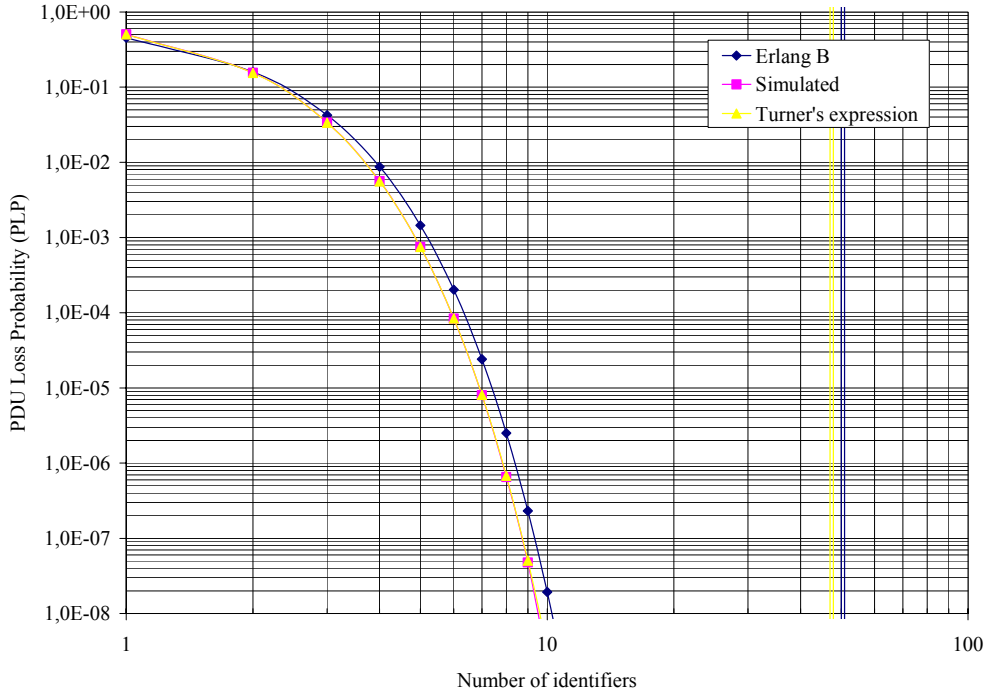

Figure 43. Comparison of PLP curves obtained in three different ways. Parameters: PCR=150Mbps, N=250 sources, rest of parameters same as in reference scenario

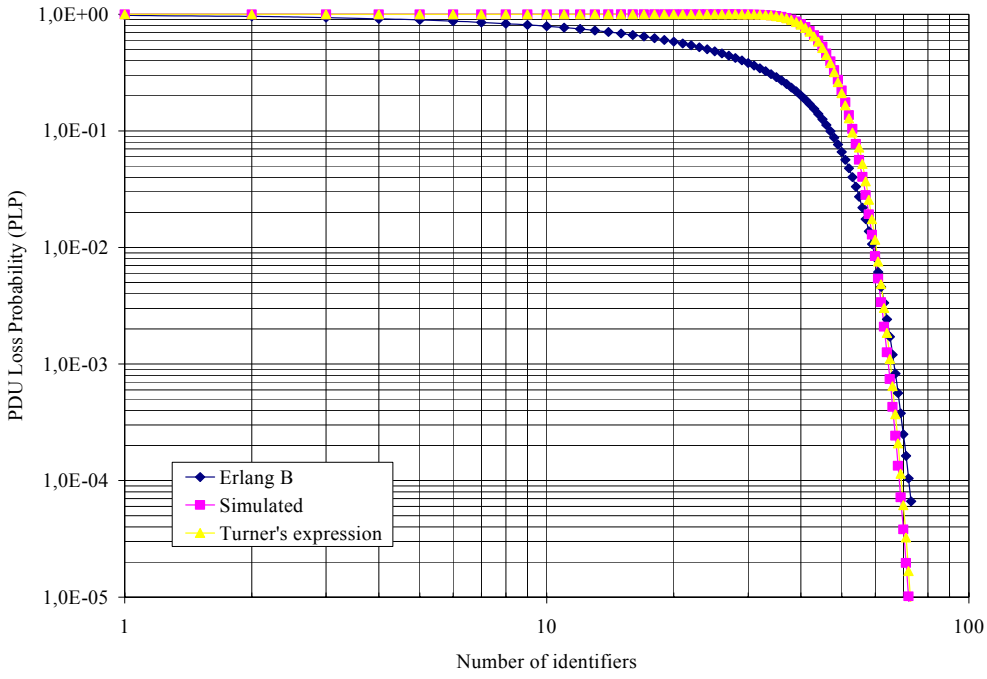

Figure 44. Comparison of PLP curves obtained in three different ways. Parameters: PCR=2Mbps, Cells per PDU=10, N=200 sources, rest of parameters same as in reference scenario

Comparisons with more than one parameter varied with respect to the reference scenario also lead to the same observations, as is the case for Figure 44, where PCR=2Mbps and PDU length=10 cells.

In light of the results obtained with the Erlang-B expression, it may seem reasonable to state that for the scenario under consideration and for our dimensioning process, the approximation of the PLP curve by means of the Erlang-B expression provides us a general and theoretical way to dimension the multiplexing ID field of the CVC connection given the traffic parameters of the source.

6.4 Summary of main results

Results show the advantages of allowing PDUs of a multicast connection to share the multiplexing identifiers as well as the interest in being able to negotiate its size.

It could also be observed that curves obtained through Turner's expression showed the same behavior as those obtained through simulation. However, one of its parameters is

not easily derived from traffic parameters of the sources, and thus, it makes it hard to use it at connection establishment.

On the other hand, Erlang-B expressions provide very similar results to those obtained through simulation for large groups. Furthermore, its parameters may be derived from those characterizing the sources, and thus, it is a good option for dimensioning the identifier at connection establishment.

Chapter 7

ARCHITECTURE OF THE COMPOUND VC LABEL SWITCH ROUTER

Once the proposal for provisioning Compound VC communications has been evaluated, our next step is to study the implementation complexity of an ATM switch or ATM LSR that provides such functionality. The goal is to determine if the additional complexity is worth it. As this is a relatively independent chapter from the rest of the thesis, it has its own structure. First, some background on ATM switches and MPLS ATM LSR architectures is presented. Particular emphasis is given to how these architectures support multicast forwarding. After that, two proposals for offering Compound VC communications are presented with their advantages and drawbacks. The required modifications of current switches is also discussed. One of the solutions solves the multipoint-to-multipoint provisioning problem with minor modifications.

7.1 Introduction

The previous chapter served to evaluate the Compound VC mechanism and to observe the benefits that such mechanism could provide in environments where multicasting is required. Thus, and as a continuation of this work, this chapter deals with the implementation issues of our proposal with the goal of studying the additional complexity of a CVC switch (or ATM LSR) with respect to current unicast switches and with respect to multicast switches. Bearing this in mind, the architecture of state-of-the-art and CVC switches is compared. Architecture should be understood in this context as the description of the functionality of each of the building blocks of these switches. However, our main focus is on ATM layer processing, and thus, the blocks that are assigned ATM-level forwarding functions are our main concern. The operation of such blocks is described by means of examples that illustrate the problems to solve for offering multicasting.

In the context of ATM switch design, multicasting usually refers to allowing one cell in an input module (IM) to be forwarded to multiple output modules (OMs), i.e. it follows a one-to-many scheme. Though this is not our definition of multicasting (see section 2.1.1), we will use such functionality of switches in the literature to provide many-to-many forwarding schemes, which is the most distinctive characteristic of CVC switches.

Two options to provide CVC communications are studied. Both have their advantages and drawbacks, but one of them provides the multipoint-to-multipoint functionality with a slight increase in the complexity of current unicast switches. Besides, notice that switches that implement other native ATM multicasting proposals also require modifications in a higher or lesser degree. Therefore, CVC switches provide the multicast service and the advantages of CVC.

This chapter also explains the similarities in the architecture of ATM switches and MPLS LSRs, and thus proposes functional blocks that provide multipoint-to-multipoint capabilities in MPLS networks by means of CVC-capable MPLS ATM LSRs. Some of the issues explained in this chapter were introduced in [Mangues 02].

As for the organization of the chapter, it is quite independent from the previous ones in the sense that it has its own background, discussion, and implementation proposal. It is organized as follows. First, it focuses on the functional blocks of ATM switches. Next,

MPLS ATM LSRs are studied. After that, the Compound VC switch and its two possible implementations is presented. The final sections of the chapter are devoted to discuss the adaptation of one of these implementations in an MPLS ATM LSR.

7.2 ATM switches

The architecture of an ATM switch is described according to the partition in functional blocks introduced in [Chen 95], and thus, this section is mainly based on this work. Figure 45 shows the relation among all these blocks.

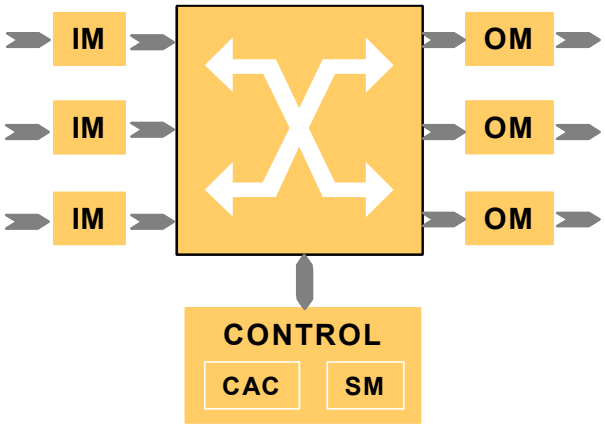

Figure 45. Generic switch block diagram

We may start by roughly describing the functional blocks of the ATM switch represented in Figure 45. The Connection Admission Control (CAC) block is in charge of dealing with the signaling involved in the establishment, maintenance, and termination of a connection. The negotiation of the traffic contract with the customer and, according to it, the determination of the parameters for the Usage Parameter Control (UPC) are also a responsibility of the CAC block.

The management of the switch is assigned to the System Management (SM) block. Some of its main responsibilities are: Operation and Maintenance (OAM) cell handling for network monitoring and debugging, configuration of switch components, traffic management, and resource usage measurement for customer billing.

These two functional blocks, though represented as centralized blocks in Figure 45, may have their functionality distributed among the Input Modules and Output Modules.

In this way, the risk of bottleneck creation is reduced at the expense of a more complex switch architecture.

The input module (IM) receives incoming cells from the physical medium, maps their VPI/VCI, and prepares them to be routed by the switch fabric. The Cell Switch Fabric (CSF) takes these cells and forwards them to the appropriate Output Module (or modules). And finally, the Output Module (OM) buffers the cells coming from the IMs and sends them to the output link after having prepared them appropriately.

At this point, some notational comments are in order. Though the expression *ATM switch* is usually found in the literature as a synonym of cell-switch fabric (the central block in Figure 45), this is not the case in this thesis. An ATM switch is composed of all the blocks represented in the figure and does not refer to just one of its parts. Additionally, throughout this thesis *module* and *port* are used as synonyms.

From the functional description above, the reader may have noticed that there are three kinds of cell flows across an ATM switch, namely user data, signaling, and management. The latter two may be forwarded by the CSF in the same way as user data cells, though towards special internal ports, or may have different paths inside the switch that directly connect the IMs and OMs with the CAC and SM blocks. Signaling and management cells usually are not transparent to the switch in the sense that the payload is examined and processed by switches in their path. On the other hand, user data cells are transparent, and thus, just the cell header fields are examined to determine the output port towards which the cell must be forwarded. In this thesis, we focus on user data cell flows, therefore, much attention is devoted to their path inside a switch, i.e. IM, CSF, and OM. A more detailed explanation of each of these blocks is given in the following sections.

7.2.1 Input Module (IM)

The functional diagram of an Input Module is presented in Figure 46. Its functions may be separated into two main groups, namely physical layer functions and ATM layer functions. The former are in charge of optical to electrical conversion, if needed, and the processing of SONET/SDH frames, in case it is the transport network used under ATM.

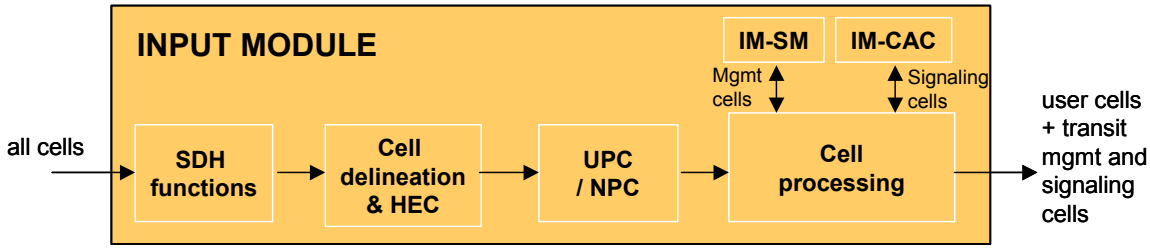

Figure 46. Block diagram of an Input Module

Prior to entering in the details of ATM-level forwarding, let us introduce all the blocks represented in Figure 46. The *SDH functions* and the *Cell delineation & HEC* blocks are responsible for the physical layer processing. The cell flow entering the IM is extracted from the SDH virtual containers, and eventually, cells are delimited and delivered to the ATM-level blocks. There are three types of ATM cells, namely user data, management, and signaling cells. The latter two are processed by the *IM-SM* and the *IM-CAC* blocks, respectively. Depending on the switch architecture, blocks related with system management and signaling are centralized, like in Figure 45, or are distributed through the parts of the switch. Therefore, the IM-SM and IM-CAC may not always be present.

As for the ATM user data plane, the IM has two main blocks. The *Usage Parameter Control/Network Parameter Control (UPC/NPC)* is in charge of verifying if the customer is fulfilling the contract and taking the corresponding actions in case it is not. Once this checkpoint is successfully traversed by the cell, it arrives at the cell processing block, where the core of the forwarding functions take place.

As far as user data cells are concerned, the cell processing block is composed of two main blocks. The VP/VC database and the header translation block. The database is a lookup table with VPI/VCI translation information. A basic forwarding table is presented in Table 9.

Therefore, when a cell arrives at the header translation block, the entry corresponding to the input VPI and VCI is looked up in the database, and the new output VPI and VCI values to write in the header are obtained. This entry also provides the output port towards which the cell must be forwarded by the switch fabric. This latter information is usually appended to the cell and helps the switch fabric in routing the cell. Additionally,

each entry in the database may also contain other information related to a particular virtual channel, e.g. cell-delay and cell-loss tolerances.

Table 9. Example of VPI/VCI translation table in an IM (values in hexadecimal)

Input		Output		
VPI	VCI	VPI	VCI	OM
2	8340	3	AF28	1
3	A341	4	3329	3
5	E312	A	522A	3
A	F3F3	B	AF2B	4
…		…		

Apart from the tag containing the output port, additional information, which is equally internal to the switch, may also be appended to the cell. It may be used for routing or switch performance monitoring. An example of the first case is a tag indicating the delay priority of the cell, which would allow to choose a path that allowed to accomplish the delay requirements. As for the second case, some examples are a timestamp and a cell sequence number that would allow to measure the transit time of the cell through the switch.

All the previous discussion assumes unicast forwarding, but for the purposes of this thesis, multicasting is a fundamental functionality. How multicasting is provided depends on the type of switch fabric. From the point of view of the IM, it may be offered by means of a tag carrying either the multicast connection identifier or a list of the OMs to reach, that is used by the CSF to make multiple copies of the cell and make all these copies get to the right output ports.

7.2.2 Cell Switch Fabric (CSF)

The main function of the Cell Switch Fabric CSF), from the user data plane perspective, is to make cells arriving at IMs get to the correct OMs. Depending on the requirements imposed over the switch, it may have many other functions [Chen 95]:

- Cell buffering
- Traffic concentration and multiplexing

- Redundancy for fault tolerance
- Cell scheduling based on delay priorities
- Selective cell discarding based on loss priorities
- Congestion monitoring and activation
- And the one we are more interested in, multicasting.

The way multicasting is offered depends on the kind of CSF. [Chen 95] differentiates four main types, namely shared memory, shared medium, full interconnection, and space division.

In the *shared memory* approach, there is a pool of memory where cells are written sequentially. As they arrive at IMs, their headers and routing tags are passed to the memory controller, which is in charge of deciding the order in which they are going to be delivered to OMs. There are two options to offer multicasting. In the first one, cells could be replicated before being written to the memory, and then, each OM reads one of these copies. This solution requires more memory than the second one, in which IMs just write one copy of the multicast cell into the memory and this cell is read many times (one for each OM where the cell should be forwarded to). Both options require some additional control circuitry to provide multicasting capabilities.

In the *shared medium* approach, the core of the CSF is a bus or ring that is shared by all IMs and OMs, i.e. all cell transfers between IMs and OMs is done through the same communication channel, and thus, some arbitration mechanism for access control is required. In this approach, OMs have address filters which examine the routing tag attached to the cell and determine if the cell must be forwarded through that OM. As a consequence of this operation, multicasting may be offered in a natural way in this switching schemes because all OMs see the cells in the bus. Therefore, if there exists a multicast address which is carried in the routing tag and the filters in the OMs are adapted to support multicast addresses, multicasting may be offered with minor changes to the CSF.

In the *fully interconnected* approach, there are as many buses as OMs. When a cell arrives at an IM, it is forwarded in parallel to all buses. The address filters are in charge of

selecting the cells that are destined to each of the OMs in the same way they did in the previous approach. Therefore, multicasting is offered by allowing address filters to support multicast addresses.

Finally, in the *space division* approach, the CSF is composed of stages of small switching modules. Each of these modules has a small number of inputs and outputs, e.g. two, and is in charge of routing the cell at the input to the upper or lower output according to the value of a control bit. These small modules are grouped to form switching modules with a greater number of inputs and outputs. These kinds of switches have some advantages, e.g. cells are routed in parallel, all elements work at the same speed, and they have simple hardware implementations. On the other hand, some mechanisms to avoid internal blocking may be required to prevent two cells being routed in parallel through the CSF to conflict. As for multicasting, it is not as easy as in the two previous approaches, though it is feasible. There are two main actions to take for multicasting, namely cell duplication and cell routing. In some cases both functions are mixed in the elements composing the CSF, which requires more state information or memory in the elements, depending on the approach. In some other cases, there may be a functional block in charge of cell replication, called copy network, which is separated from the block in charge of routing the cells. With respect to the tags, the most common approach is to carry multicast addresses, which are used by CSF elements to make all the copies of the input cell get the all the OMs.

All the schemes for offering multicasting described above permit the transfer of one cell in an IM to many OMs. We benefit from this functionality to provide the ability to transfer many cells coming from different IMs to many OMs, thus allowing the creation of shared trees.

7.2.3 Output Module (OM)

The function of the OM is to prepare the cell for transmission through the medium. In general, for unicast switches, the operations carried out in the OM are simpler than those in the IM, and they roughly carries out the reverse operations. According to [Chen 95] the functions of an OM are:

- Removal and processing of internal tags
- Possible translation of the VPI/VCI values
- HEC field generation and inclusion into cell headers
- Traffic shaping
- Possible mixing of signaling and management cells with outgoing user data cells
- Cell rate decoupling (adding empty cells)
- Mapping cells into SDH payloads
- Generation of SDH overhead
- Conversion of the digital bit stream into optical signal

A functional block diagram to handle all these functions is presented in Figure 47.

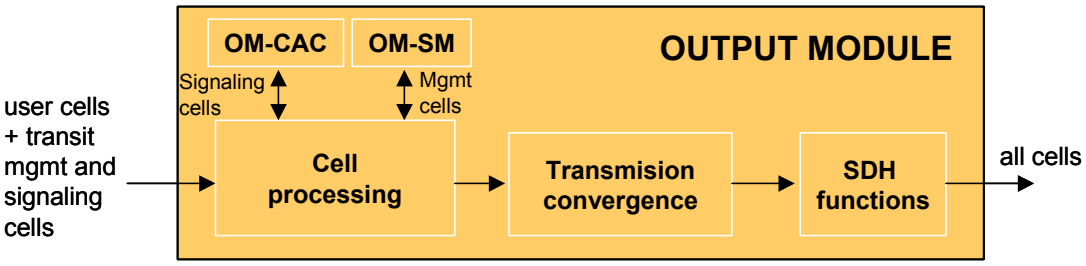

Figure 47. Functional diagram of the output module

The same comments with respect to IM-SM and IM-CAC in the IM section apply now for the *OM-SM* and *OM-CAC* blocks. As explained above, the processing of management and signaling cells may be distributed or centralized. If centralized, these two blocks are not present in the OM and management and signaling cells are delivered to an internal port to which the centralized block is attached.

The *transmission convergence* and *SDH functions* block are in charge of physical layer functions, and in that sense, prepare cells to be forwarded through the medium, by means of SDH frames in this case.

As this thesis focuses on ATM layer processing, the emphasis will be in the first points of the list above, which are concentrated in the cell processing block. The functionality of this block is represented in Figure 48. The first two blocks are in charge of adding the

management and signaling cells coming from the SM and CAC blocks to the cell flow. As far as user data cells are concerned, they are processed at the last block.

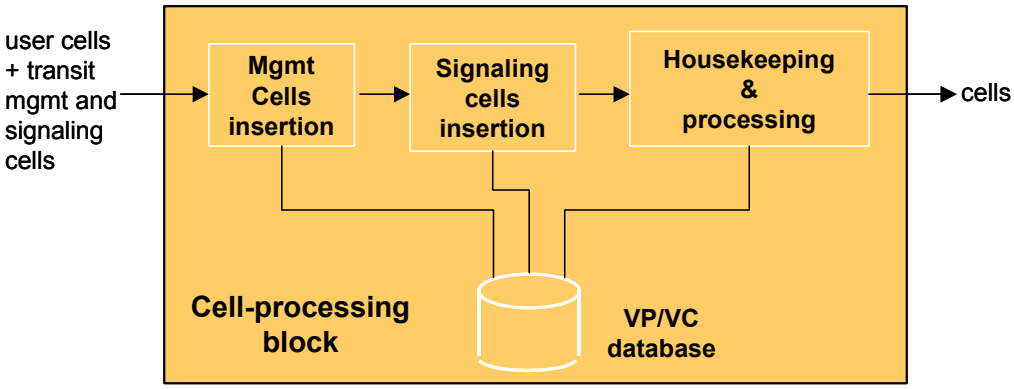

Figure 48. Functional diagram of the cell-processing block

An outgoing cell carries an internal tag when it exits the CSF. This internal tag may serve various purposes. The main one is to allow the routing of the cell through the CSF. If the cell is unicast, this tag may be removed and no extra processing is required with the tag because the VPI/VCI mapping was already carried out in the IM. On the other hand, for multicast cells, this mapping must be carried out in the OM. Otherwise, all cells at different OMs would end up by having the same output VPI/VCI value, which may not be acceptable in the general case. Therefore, the OM is in charge of mapping the multicast connection identifier in the internal tag to the outgoing VPI/VCI values.

The internal tag may also be used for housekeeping purposes, i.e. measuring the internal performance of the switch. In this case, the cell processing block handles such tasks. For instance, timestamps and sequence numbers may be inserted in the IM to calculate the time elapsed by each cell in the switch and, in this way, the OM may keep track of this delay. Once all these processing ends, the tag is removed.

As for the mapping, the processing block requests some information to the VP/VC database, which contains an entry for each output VPI/VCI. This entry associates a multicast connection identifier with the corresponding outgoing VPI/VCI values. It may also contain additional information for each VCC, e.g. housekeeping information, traffic-rate information, or egress cell counts. According to the registered traffic-rate information, traffic shaping may also be performed by the OM for congestion control.

Once all this processing is finished, other functions are performed, like HEC generation, cell insertion in SDH payloads, SDH overhead generation, and, in general, adaptation of the outgoing cell streams to be transmitted through the output link.

7.2.4 Example of multipoint-to-multipoint switch: Store-and-Forward (or VC-merge)

The main difference in a Store-and-Forward (SF) switch (or VC Merge switch) with respect to conventional switches is in the Output Module. All the required buffering management is carried out there. Figure 49 presents a block diagram of such an output module. Only ATM cell forwarding parts are represented. SONET/SDH operations, signaling and management cell insertion, etc. are not represented, as they are equivalent to those presented above.

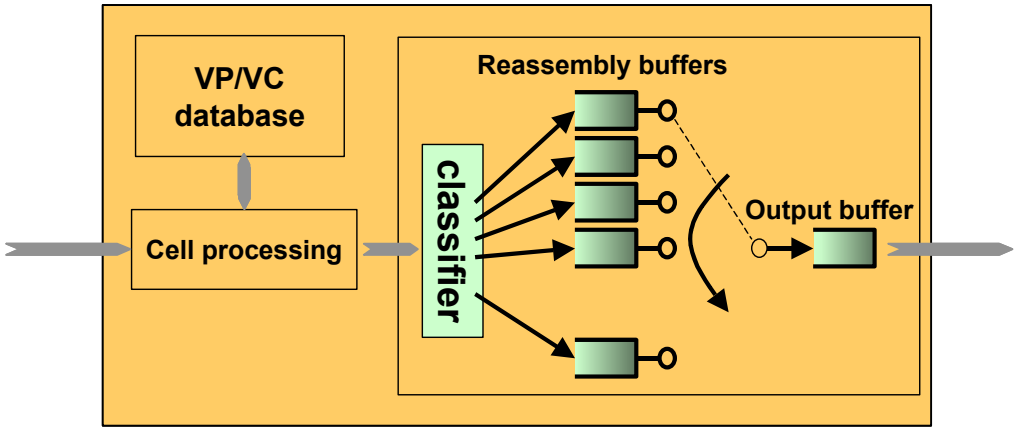

Figure 49. ATM level processing of user data cells at an Output module of a Store-and-forward (or VC merge) switch

When cells arrive at the OM after having crossed the switch fabric, they are classified and put into their reassembly buffer according to the tuple (input module, VCI). Notice that the IM must add an internal tag to allow the OM to differentiate between cells coming from different IMs with the same VCI [Widjaja 99]. The IM number is carried in an internal tag. Once all the cells of a PDU arrive, they are moved in an atomic manner from the reassembly buffer to the output port from where they are transmitted through the output link.

7.3 MPLS ATM Label Switch Routers

In MPLS terminology [Rosen 01], there are two main kinds of nodes, namely Label Switching Routers (LSRs) and MPLS edge nodes, often referred to as Label Edge Routers (LERs). The latter are in charge of connecting non-MPLS domains with MPLS domains. Their main functions are related to the classification of packets into Forwarding Equivalence Classes (FECs) and the attachment of the corresponding label to each packet entering the MPLS domain. As they interact with conventional layer-3 networks, they must implement mechanisms for both longest prefix match forwarding and label switching forwarding. As for LSRs, they are in charge of forwarding inside MPLS domains, and thus, they just forward packets based on their labels.

Focusing on the LERs, their forwarding functions are carried out in a block that is usually referred to as Forwarding Engine (FE). There are two options in the way the transition from ATM equipment to MPLS equipment may be targeted in these nodes [Lee 99]. The first one is to add a centralized FE to an ATM switch. That is, one port of the switch is connected to the FE and the rest act as a usual ATM switch. All the traffic that should be label switched is forwarded to the FE, which changes the label and forwards the packet to the corresponding output port of the ATM switch. In this way, the migration path is relatively simple, because the difference between an ATM switch and an MPLS ATM LSR is the absence or presence of this single new component, respectively. However, this centralized FE may become a bottleneck in case the LER receives high volumes of traffic to be label switched. Therefore, this scheme does not scale. To satisfy the scalability requirements, [Lee 99] advocates for a second option: the distributed FE configuration, which matches the ATM switch forwarding architecture introduced in section 7.2 in terms of the distribution of the processing among all the modules.

Once in the MPLS domain, the forwarding approach of MPLS follows the principle “switch when you can, route when you must.” It is based on the old idea of virtual circuits, and thus, the processing of packets inside a node consists of determining the value of the output virtual circuit identifier based on a table lookup operation that is indexed by the input interface and the value of the input virtual circuit identifier. Therefore, these identifiers are locally remapped at each node in the network. In MPLS terminology, the virtual circuit is called *Label Switched Path (LSP)* and the virtual circuit

identifier is called *label*. This is exactly the same way ATM switches process cells. As a consequence, MPLS LSRs and ATM switches show many similarities.

As a matter of fact, the development of ATM switches and their forwarding efficiency triggered some initiatives to provide this same efficiency to IP packets. The early approaches to what was called IP switching opted for maintaining the switching blocks of ATM switches and just changed the software of the nodes (see section 2.3.3). IETF compiled all IP switching proposals and refined them to finally generate the Multiprotocol Label Switching Architecture [Rosen 01], which generalizes the switching principle for media other than just ATM.

However, and due to the wide deployment of ATM equipment in the networks of carriers, ISPs, and companies, ATM is still a common technology. In this chapter, we will mainly focus on the comparison of ATM switches with MPLS LSRs that use ATM as the link layer technology, i.e. MPLS ATM LSRs. And more particularly, this section is mainly devoted to explain the forwarding characteristics of MPLS ATM LSRs. As they may be understood as un upgrade of existing ATM switches, a lot of parallelism is established with section 7.2. As the focus is on forwarding and not on control, the control blocks of the architecture are not dealt with.

The block diagram of the MPLS ATM LSR is the same as the one presented above for the ATM switch (Figure 45). Let us begin with the lower layer processing. The physical layer of an MPLS ATM LSR does exactly the same processing as in a common switch. Therefore, in case of using SDH as transport layer at the physical layer, the main functions are electrical-optical conversion, SDH overhead processing, and cell-rate decoupling (i.e. discarding or addition of empty cells).

Though in an MPLS ATM LSR, cells are still used to send the information through the network, the ATM layer processing is changed for an equivalent processing at the MPLS layer, whose principles of operation are basically the same (see section 2.3.3). If we take a look at the ATM layer forwarding functions described in previous sections, we see that some of them need to be slightly modified, but the basic functionality is the same. These functions were mainly concentrated in the cell processing block of both Input and Output

Modules. The following paragraphs establish a parallelism between the ATM switch functional blocks and the MPLS ATM LSR blocks.

As for the Input Module, the following comments apply with respect to what has been explained in section 7.2.1. The core of the forwarding procedure is located at the cell processing block, whose basic function is the same, but in this case, the VP/VC database is actually a Label database (Label Information Base, or LIB) and the header translation block instead of changing the VPI/VCI values changes the label that is carried in the VPI/VCI fields of the ATM cell header. Therefore, an example of how a forwarding table contained in the LIB may look like is presented in Table 10.

Table 10. Example of LIB table in an IM (values in hexadecimal)

Input label	Output label	OM
1132	1F38	5
3A4F	312A	3
A41A	5C9A	2
B62E	BF2B	2
…	...	

The only difference with respect to the table in section 7.2.1 is that the VPI+VCI columns are merged into one Label column. The rest of the additional information at each entry, like internal tags, are the same, and are equally required to allow the CSF to work properly.

The CSF of the MPLS ATM LSR is exactly the same as that of an ATM switch, as ATM LSRs use, by definition, ATM switches as their internal core. In fact, most high performance IP routers (not just MPLS ATM LSRs) internally divide their packets into chunks of bits of fixed-size due to the advantages in resource management it implies [Metz 98], and as a consequence, may benefit from the research on ATM CSFs.

The Output Module is also very similar. When the cell arrives at the OM, the internal tag that allowed correct routing of the cell inside the CSF is removed and the other tags are processed. For instance, there may be internal tags for housekeeping purposes, and thus, they are removed after the housekeeping information is obtained. The tag that is

used to allow multicasting of cells may require further processing in the same way as in the ATM switch. Therefore, additional label lookups based on the contents of this tag may be required. Once more, the role that VPI/VCI fields played in the ATM switch is now played by the label contained in those fields.

In conclusion, the main changes with respect to a conventional unicast ATM switch come from the provisioning of the multicasting functionality. These changes are also required in ATM switches. Therefore, the change from ATM to MPLS technology is not as important as the change towards multicasting as far as forwarding is concerned.

The previous paragraphs showed the relatively simple migration path towards MPLS. This latter advantages add to those of using ATM fixed-size cells, e.g. benefiting from the installed base of ATM equipment, simpler switching, and simpler resource allocation. Examples of implementations that add the required additional blocks to transform an ATM switch into an MPLS ATM LSR to benefit from the previous advantages may be found in a number of papers related with the implementation of Electronics and Telecommunications Research Institute (ETRI) HAN-bit ACE 64 ([Lee 99], [Kang 00], [You 00]) and other similar proposals [Lee 01]. Most of these papers are devoted to explain the architecture of an MPLS ATM Label Edge Router (LER), that is, the kind of MPLS nodes that are found at the ingress of the network, and not MPLS ATM LSRs because ATM switches show an easy conversion into LSRs, as explained above. The main complexity in LERs resides in the classification of traffic into FECs, which involves IP header processing. Thus, the additional complexity does not come from the label swapping paradigm of MPLS that is well supported by ATM equipment.

7.4 The Compound VC switch

Having seen the similarities between ATM switches and MPLS ATM LSRs, we now propose the architecture of a switch, that may as well be used in LSRs, that provides multipoint-to-multipoint capabilities. We refer to this switch as the CVC switch.

This discussion mainly focuses on the forwarding of user data cells. Therefore, most attention will be on the IM-CSF-OM path. And most particularly, we study the requirements of the IM and the OM for ATM-layer processing. The only requirement we

impose over the CSF is that it be capable of forwarding one input cell to many output ports. The physical layer processing is the same as described above.

There are two options for the implementation of the CVC switch, namely the Longest Prefix Match (LPM) and the 2-Mappings (2MAP). The first one allows to treat the group as a whole by just having one entry for any given group in the CVCID mapping table of the IM. This implementation involves LPM lookup operations because the CVC ID is variable in length. LPM lookups are costly to implement in hardware. On the other hand, in the 2MAP implementation, the switch must carry out two mappings of the VCI field of each cell. But these mappings are carried out with indexing table lookup operations. This thesis focuses on 2MAP because it is simple and it may be implemented with state-of-the-art ATM hardware.

As a general remark, the deployment of CVC is only possible if all switches in the network are capable of mapping multiple VCs to the same output logical queue. Otherwise, it is possible that cells belonging to the same CVC connection were enqueued in different queues. As a consequence, and depending on the scheduling applied in that switch, the order of cells may be altered, and thus, one of the main characteristics of the connection-oriented philosophy of ATM may be broken. This remark was commented in [Venkateswaran 98] with respect to DMVC but it also applies for all Multiple VC switch strategies, and thus, also for CVC.

Another characteristic that is common to both options and also to any multicast switch is that two VPI/VCI lookup operations are required, one at the IM and another one at the OM [Chen 95].

With these characteristics in mind, which are common to the LPM and the 2MAP implementations, the following sections describe how a switch provides the multipoint-to-multipoint capabilities in both cases.

7.4.1 Longest Prefix Match (LPM) implementation

The underlying idea of this approach is to switch all cells belonging to the same Compound VC using a single entry in the tables with the aim of having one entry per

multicast connection. As the connection ID (CVCID) has a variable length, this involves longest prefix match lookup operations.

Therefore, if we focus on the ATM layer processing of user data cells, what we describe below is the processing carried out in the cell processing block of the IM. In particular, it deals with the header translation function that also involves the VP/VC database.

7.4.1.1 Input Module

Inside the Input Module, the header translation block is in charge of triggering the lookup operation into the VP/VC database, which contains tables like those depicted in Table 11.

Table 11. Input Module lookup table in the LPM implementation (values in hexadecimal)

Input			Output
VPI	CVCID	Mask	Multicast Connection Identifier (MCI)
3	7340	FFF0	35
4	2348	FFF8	54
A	AE20	FFE0	A1
E	C1A0	FFF0	DD
…			…

A new column in the switching table is required to consider a mask, but table size is not globally increased as there is just one entry for the whole group. Actually, the four entries in this table correspond to one Compound VC connection each. The mask determines the portion of the VCI that identifies the compound VC. For instance, a mask with value 0xFFF8 means that the size of the CVCID is 13 bits for that multicast connection. Thus, the 3 remaining bits in the VCI are used as PDU ID.

Back to the processing, and as a result of the header translation operation, the cell is attached an internal tag carrying the multicast connection identifier (MCI). This tag is used by the CSF to make all copies of the multicast cell get to the output ports. The actual size of this tag depends on the implementation of each switch. Another tag carrying the IM number is also attached and will be processed by the OM. There is no change in the VPI/VCI values in the IM.

7.4.1.2 Output Module

Once the multicast cell gets to the OM, the cell eventually arrives at the cell processing block, where another table lookup operation is triggered. During this operation, the VP/VC database is queried by using the multicast connection identifier as index to the table. Table 12 shows how such a table may look like.

Table 12. Output Module lookup table in the LPM implementation

Input	Output			
MCI	VPI	CVCID	Mask	PDUID table pointer
54	A4	EDC8	FFF8	@1
CC	A	3A40	FFFE	@2
	…			

In this process, the new output VPI and CVCID values are obtained, and they are written into the header of the cell. Following the example in the tables, we see that there are two multicast connections (54 and CC). Those cells with MCI=54, disregarding the IM from where they came, have as CVCID 1110110111001, that is, the first 13 bits of the result of AND (EDC8, FFF8).

The remaining part of the VCI (the PDU multiplexing ID) is handled in a per-connection basis by means of tables like the one in Table 13. A pointer to one such table is obtained from the last column of the OM table.

The PDU ID mapping table is required to keep track of the identifiers in use, that cannot be assigned to cells that do not belong to the same PDU, and those that are free, that are assigned to the cells of newly arriving PDUs. For this purpose, a new mapping between PDU IDs is required. Finally, in case all PDU IDs of this Compound VC are simultaneously allocated, the new arriving PDUs will have its PDU ID marked as lost, which corresponds to a value of 1 in the last column of the table. Therefore, all remaining cells will be discarded.

Table 13 corresponds to a connection with 3 bits in the VCI being used as PDUID, and thus there may be as much as 8 simultaneous PDUs being forwarded. Notice that for this multicast connection, there are three IMs from where cells arrive (1,3, and 4). The tag

containing the IM number is required to differentiate cells coming from different IMs with the same CVCID + PDUID, and thus, avoid potential cell-interleaving.

Table 13. PDU ID mapping table in the LPM implementation for a given MCI with PDU ID length of 3 bits (8 identifiers).

Input		Output	
IM	PDUID	PDUID	Lost
1	1	-	1
1	2	0	0
1	3	1	0
1	7	2	0
3	2	3	0
3	3	4	0
3	4	5	0
4	2	6	0
4	5	7	0

Notice also that in this implementation, the OM is more complex than that of a conventional multicast switch due to table management, and particularly, due to PDU ID table management. In the following section, another implementation option that tries to simplify cell processing is presented.

7.4.2 Two-mappings (2MAP) implementation

The main idea under this implementation is to simplify the hardware required to offer multipoint-to-multipoint CVC communications with respect to the previous one. The main problem of the above proposal was that it involved longest prefix match operations due to the variable size of the CVCID. In this implementation, both variable-size identifiers (CVCID and PDU ID) are locally remapped at each switch by performing the mapping of a fixed-size field (the VCI field) twice. This does not add extra complexity to the switch with respect to conventional multicast (i.e. point-to-multipoint) switches, because, as stated in [Chen 95], multicast switches usually require two lookup operations. The first one is performed at the IM to obtain the output ports to which the cell must be forwarded, and the second one, at the OM, to determine the VPI/VCI values to write in the header of the outgoing cell.

The following sections explain by means of an example how each of these mappings is carried out in a 2MAP CVC switch. The sample scenario used to describe the operation of the 2MAP CVC switch is represented in Figure 50. The operation of the IM, where CVC ID mapping is carried out, is first explained. After that, it is the turn for the Dynamic VCI Assignment (DVA) module, which is located at the OM and is in charge of PDU ID mapping.

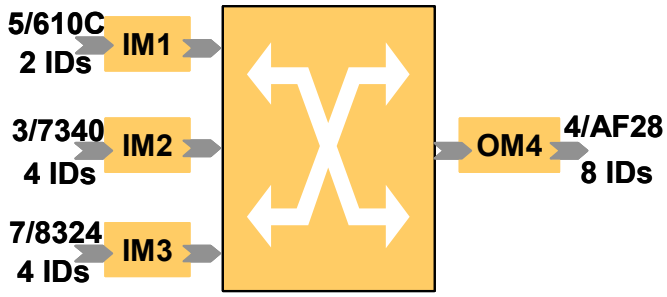

Figure 50. Cell forwarding in a CVC switch.

7.4.2.1 Input Module – CVC ID mapping

The IM is in charge of the mapping of the CVC ID. This mapping, for a given CVC connection, is determined at connection establishment. That is, the entries in the CVC ID mapping table for a given group are filled in when the connection is established and are removed from the table when the multicast connection ends. The number of entries in the mapping table of a given IM corresponding to a single CVC connection is equal to the number of IDs used in the input link to that IM. In our example (Figure 50), 2 IDs (and thus, 2 entries in the table) for IM 1, 4 IDs (4 entries) for IM 2, and 4 IDs for IM 3.

The mapping tables that may be found in the IMs of the example are shown below. Table 14, Table 15, and Table 16 respectively describe the CVCID mapping tables at IM1, IM2, and IM3 for a given CVC connection. For example, the traffic belonging to this group enters input module 2 with VPI=3, CVCID=01110011010000 (these bits correspond to the more significant bits of the VCI field). As we have two bits assigned to the PDU ID, there are four possible identifiers. That is why there are four inputs in Table 15. The same considerations apply for the rest of the tables of the input modules. For all tables, just entries corresponding to the CVC connection in Figure 50 are shown.

Table 14. CVC ID mapping table at input module 1

Input		Output		
VPI	VCI	VPI	VCI	OM
5	610C	4	AF28	4
5	610D	4	AF29	4

Table 15. CVC ID mapping table at input module 2

Input		Output		
VPI	VCI	VPI	VCI	OM
3	7340	4	AF28	4
3	7341	4	AF29	4
3	7342	4	AF2A	4
3	7343	4	AF2B	4

Table 16. CVC ID mapping table at input module 3

Input		Output		
VPI	VCI	VPI	VCI	OM
7	8324	4	AF28	4
7	8325	4	AF29	4
7	8326	4	AF2A	4
7	8327	4	AF2B	4

These entries are used to map the CVC ID field of the incoming cells and to forward them to the appropriate output port/s. The 2MAP implementation treats the entries in the IM table as if they were normal VCIs. Thus, state-of-the-art ATM hardware could be used. This is in contrast with the LPM implementation, which just requires one entry in the table for each group. But the price paid by LPM is major changes in ATM hardware.

Notice that the PDU ID is also mapped jointly with the CVCID when the whole VCI is changed. But the PDU ID will only be paid attention in the OM.

The range of VCIs allocated to the output link will determine the CVCID and PDU ID values to which incoming cells are mapped in the IM. In our example, the output VCIs allocated for the group at OM 4 are comprised in the range 0xAF28 to 0xAF2F (see Table 17). This range is chosen at connection establishment and the OM sends to each IM involved in the forwarding of the CVC connection a number of VCIs inside this range equal to the number of input IDs to that IM. These VCIs will be used to fill in the output

VCI column of the CVC ID mapping table. For instance, OM 4 sends IDs 0xAF28 and 0xAF29 to IM1, which is assigned 2 IDs

Once the CVC ID mapping is done, a routing tag is added to the cell. This tag will allow the switch fabric to route the cell to the correct output port. If there is a single destination port, the tag will directly contain the output port number. But if there are multiple destination ports, this tag will carry a multicast address (or Multicast Connection Identifier) that will allow the multicast-capable switch fabric to duplicate and route the cell to the corresponding ports.

A second tag is also added to the cell. It contains the IM through which the cell arrived to the switch. This field is required in case that cells coming from different IMs with the same VCI, and thus, the same CVCID and PDU ID, are forwarded to the same OM. In this way, cell-interleaving is avoided in all cases.

Therefore, in most operations, the IM behaves exactly like a normal ATM switch. The only slight difference is the inclusion of a second information tag next to the routing tag, which is also required for store-and-forward (or VC merge) hardware [Venkateswaran 98]. This means that state-of-the-art hardware could be used in the Input Module with slight modifications with respect to unicast switches and no additional complexity with respect to some multicast capable switches.

7.4.2.2 Output Module - Dynamic VCI Assignment (DVA)

Figure 51 presents the internal structure of the cell-processing block of the Output Module of a CVC switch. Only blocks used for handling CVC cells at the ATM level are represented. Other blocks for signaling and management cell handling are not represented.

The OM is assigned the task of PDU ID mapping to avoid ID collision when traffic coming from different input ports is multiplexed. Therefore, if such multiplexing does not occur, the DVA block functionality is not required, and cells are forwarded with just CVC ID mapping.

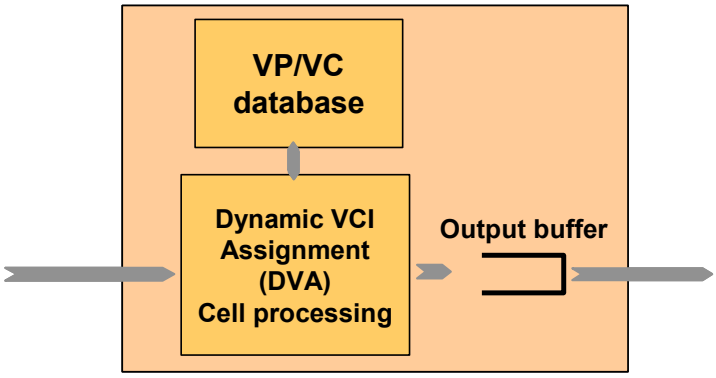

Figure 51. Cell processing block of output Module of a CVC switch

The operation of the DVA block is better described by means of Table 17, which corresponds to the PDU ID switching table at output module 4. In the example, all the cells coming from IM1, IM2, and IM3 have been given a CVCID equal to 1010111100101, and the three less significant bits in the VCI field of the OM are used as PDU ID. Thus, there are 8 IDs.

Table 17. PDUID mapping table at output module 4

Input		Output	
VCI	IM	VCI	discard
AF28	1	-	1
AF29	1	AF28	0
AF28	2	AF29	0
AF29	2	AF2A	0
AF2A	2	AF2B	0
AF2B	2	AF2C	0
AF28	3	-	1
AF29	3	AF2D	0
AF2A	3	AF2E	0
AF2B	3	AF2F	0

When a cell arrives at the OM after being routed by the switch fabric, it arrives at the cell-processing block. PDU ID mapping is carried out in the Dynamic VCI Assignment (DVA) block (Figure 51). Notice that though the PDU ID is variable in size, its mapping is carried out by changing the whole VCI (Table 17). Thus, the OM just handles fixed-length fields. Notice also that though we map the entire VCI, the only thing that is changed in the OM is the PDU ID, because the CVC ID was already changed in the IM.

Therefore, when the cell enters the cell-processing block, the DVA looks up the VP/VC database and translates the VCI field in the cell header depending on the tuple (input module, input VCI) (Table 17). The output VCI carries a unique PDU ID for that CVC connection. That is, no other PDU belonging to the same CVC connection and passing through the same output module at the same time is assigned that ID.

Once the translation is done, the internal tags are removed and the cell is stored in the output buffer, from where it is transmitted to the output link.

The actions to undertake when mapping a cell depend on the type of cell inside an AAL5-PDU (initial, middle or last). When the initial cell arrives, a free PDU ID is assigned to the PDU and the PDU ID is marked busy to avoid other PDUs from using it. But, there may be some cases in which all the identifiers of all the input modules are in use. In the example, ten simultaneous PDUs are trying to pass through OM4 (2IDs from IM1 + 4IDs from IM2 + 4IDs from IM3). As shown in Table 17, all cells of two PDUs are discarded due to having run out of output identifiers (discard column equal to 1).

If middle cells must not be discarded, once they enter the DVA block, the (input module, input VCI) tuple is searched in the table and the output VCI (which includes the PDU ID) of the matching entry is written in the cell header.

The binding between the input tuple and the output PDU ID lasts until the last AAL5 cell arrives, i.e. once the last cell arrives, the same mapping process is carried out, but after that, the output PDU ID is freed and the entry is removed from the table.

Once again, the implementation is carried out by mapping the whole VCI instead of just the PDU ID with the aim of preserving most operations of a normal ATM switch. In this way, with only slight modifications in VCI table processing, it is possible to offer multipoint-to-multipoint connections without degradation of traffic characteristics. Besides, the OM of some current multicast (point-to-multipoint) switches may require cell processing blocks to map multicast connection IDs to output VPI/VCI values [Chen 95]. Therefore, in this case, there shall not be any extra hardware apart from the one that controls the actualization of the VP/VC database each time a PDU starts or ends, which may be comparable to the management of reassembly buffers in store-and-forward (or VC-merge) switches.

Thus, and as a concluding remark, notice that in the 2MAP implementation, the switch is capable of mapping two variable length fields (CVCID and PDUID) by performing two fixed-length mapping operations. As a consequence, the CVC mechanism may be implemented with minor modifications to current ATM hardware and without loosing the flexibility of the CVC mechanism.

The example followed during the previous explanation focused on a multipoint-to-point connection. If we now focus on the multipoint-to-multipoint case, the simpler solution is to determine at connection establishment a group of adjacent VCs which have the same values in all OMs involved in the multicast connection. In this way, the same forwarding scheme as in the multipoint-to-point case applies. The only difference is that the last column of IM tables contains either a multicast connection identifier or the list of OMs to which the cell must be forwarded, depending on the particular implementation of the switch.

7.5 MPLS ATM LSR with CVC-2MAP

After what has been said in previous sections, adding the CVC functionality to an MPLS ATM LSR requires the same modifications as adding it to an ATM switch. Therefore, the role of the VPI and VCI fields are now played by the label, which is contained in the VPI+VCI fields, and thus, according to what has been explained in section 7.3, VP/VC databases are now LIBs. Apart from that, there is no additional difference as far as forwarding is concerned. The following tables and figures show how the same example in section 7.4.2 would work in an MPLS environment.

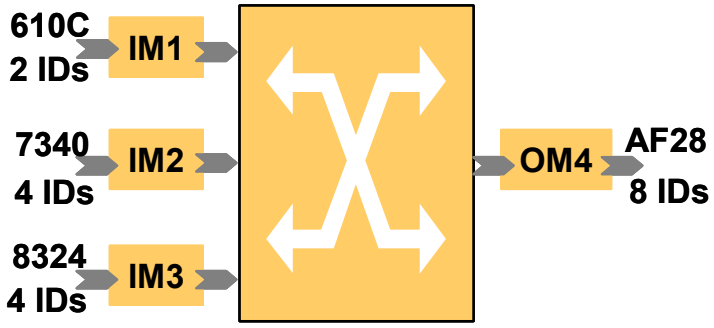

Figure 52. Example of forwarding in a MPLS ATM LSR with CVC functionality

Table 18. Example of CVC forwarding table at IM2 in the MPLS scenario of Figure 52.

Input	Output	
label	label	OM
7340	AF28	4
7341	AF29	4
7342	AF2A	4
7343	AF2B	4

Table 19. Example of CVC forwarding table at an OM in an MPLS environment

input		Output	
label	IM	label	Discard
AF28	1	-	1
AF29	1	AF28	0
AF28	2	AF29	0
AF29	2	AF2A	0
AF2A	2	AF2B	0
AF2B	2	AF2C	0
AF28	3	-	1
AF29	3	AF2D	0
AF2A	3	AF2E	0
AF2B	3	AF2F	0

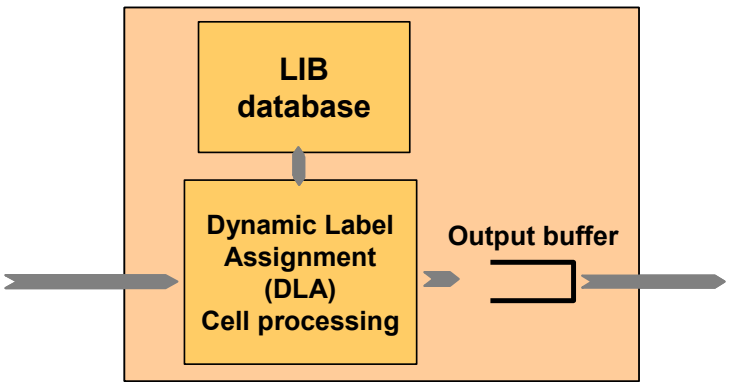

Figure 53. Block diagram of the OM of an MPLS ATM LSR with CVC functionality

As seen in the tables, four identifiers, which correspond to four labels are assigned to IM2 according to the traffic characteristics of the incoming traffic. The incoming labels are mapped into four labels that belong to the pool of outgoing labels. All this processing

takes place at the IM. Internal tags carrying the IM identifier and the multicast direction if necessary are added to the cell to help the CSF in routing the cell towards the output port/s. Once in the output module, the internal tags are removed and the cell processing block dynamically assigns labels in agreement with the information contained in the LIB of the OM. Therefore, the Dynamic Label Assignment block (DLA) works in the same way as the DVA block above.

This chapter is mainly focused on the forwarding part of CVC connections. Other aspects should also be considered. For instance, LDP should be modified to support multipoint-to-multipoint LSPs in a similar way to other multicast scenarios provided with other native ATM multicast mechanisms. However, this issue belongs to the control/signaling part, which is not the object of this thesis.

Chapter 8

CONCLUSIONS, SUMMARY, AND FUTURE WORK

This chapter serves as summary of this thesis and also presents the main conclusions that may be drawn from our work. Additionally, some lines of future research that may take our work as their basis are also presented.

8.1 Summary and Conclusions

The main goal of this thesis has been to solve the research problem stated in Chapter 4, that is, "to design and evaluate the forwarding part of a native ATM multicasting mechanism which 1) solves the cell-interleaving problem when AAL5 is used, 2) copes with most of the problems that appear in previous proposals, and 3) has application in MPLS ATM LSRs, with special emphasis on the incidence in group interactive communications."

The analysis of the previous work carried out in this area revealed that current mechanisms were applicable in some cases but not always for a kind of traffic increasing in importance as is group interactive traffic (e.g. multimedia). This lead us to propose a new native ATM multicasting mechanism, named Compound VC (CVC).

Additionally, the study of the background of this research area gave as a result the classification of the native ATM multicasting mechanisms in four main groups, namely avoid cell-interleaving, VP switching, allow multiplexing inside a VC, and multiple VC switching (Chapter 3). CVC belongs to this latter group.

CVC is particularly suited for group interactive communications. This suitability comes from its main design criteria. It allows the multiplexing of cells belonging to different Packet Data Units (PDUs) by means of a PDU multiplexing identifier, but no additional overhead is introduced. Flexibility is also a key aspect of CVC, as the size of the identifier may vary from group to group so as to adapt to varying traffic and group characteristics. It has also been designed to be an integral part of MPLS ATM Label Switch Routers (LSR). The following paragraphs develop in a bit more detail the conclusions related with these points.

Particular emphasis has been put in the comparison between CVC and store-and-forward (SF) (also known as VC merge) due to its acceptance as one of the possible mechanisms to be implemented in MPLS ATM LSRs. Section 6.2.2 showed that the throughput obtained is the same for both CVC with a given number of identifiers and SF with the same number of reassembly buffers assigned to the multicast communication.

Furthermore, CVC shows this behavior while respecting the traffic characteristics of the source and with less buffer consumption than SF.

Section 6.2.2 also presented simulation results that state the advantages of the utilization of PDU multiplexing identifiers instead of source multiplexing identifiers. The interest of the former is in that resource sharing is much bigger. As a consequence, with a small number of identifiers high throughput values are obtained and a number of sources much bigger than the number of identifiers is served. This result is further confirmed in sections 6.3.4 and 6.3.5, where much bigger groups with diverse traffic conditions are studied through simulation and analytically, which also shows the scalability of the mechanism.

Focusing on the dimensioning of the identifier, our aim has been to obtain some rules that provide the required number of identifiers for a multicast communication given an acceptable packet loss probability (PLP) to the application. If possible, this value should be lower than that caused by buffer overflow at switches or any other transmission losses. In this way, the multiplexing process would not be the limiting factor of the multicast communication. With respect to the results, the evaluation through simulation allowed us to observe that to obtain acceptable PLP values the sizes of the identifiers ranged from 4 to 7 bits in most cases. Furthermore, for these ID sizes, the PLP curve shows a linear behavior in log-log scale.

As for the analytical evaluation, the curves obtained by means of a probabilistic expression derived by Turner almost perfectly overlap in most cases with the PLP curves obtained through simulation (section 6.3.4). However, its utilization is precluded in the dimensioning process because the parameters on which it depends are not easily obtained from the parameters known by the sources at connection establishment.

On the other hand, the multiplexing identifier dimensioning problem has been matched with the circuit dimensioning problem of a telephony link for which Erlang-B expressions were derived (section 6.3.5). This approximation has proved satisfactory for our dimensioning process in most cases in the sense that PLP curves in the region of ID sizes of interest match up the behavior of those obtained through simulation. Besides, and unlike with Turner's expression, the parameters required by the Erlang-B expression may

be obtained by doing simple operations with traffic parameters that sources know. Therefore, it provides a way of dimensioning the multicast communication at connection establishment.

A further implication of the utilization of the Erlang-B expression, is that, in the homogeneous scenarios under consideration, if source and group parameter values are found so that the total traffic intensity introduced to the system is the same, the PLP curve is the same disregarding the particular characteristics of the sources.

Other observations lead us to conclude that for the values of parameters tested, PLP curves show a more significant variation when the PCR and burstiness values are changed than when the PDU length is changed.

The evaluation results also show the interest of having flexibility in the negotiation of the dimension of the multiplexing identifier, as different multicast applications may present diverse characteristics that should be handled by the mechanism without a waste of the overhead required to carry the multiplexing identifier.

We have also proposed the block-level architecture of an ATM switch, which provides multipoint-to-multipoint forwarding capabilities, implements CVC, and may serve as base of an MPLS ATM LSR (section 7.4). Its application into these kinds of nodes may help in leveraging current ATM knowledge into MPLS, as ATM is one of the more likely link-layer technologies to be used in MPLS.

There is a trade-off between allowing flexibility in the size of the identifier and the additional implementation complexity required to support it. However, it is solved in one of our implementation proposals (2MAP), which provides the multicast functionality through CVC with slight modifications to the architecture of a legacy unicast ATM switch, and shows comparable complexity to other multicast switches. Thus, current expertise in switch design may be exploited, as no major changes are required.

Furthermore, the current importance of ATM infrastructure is remarkable. It may appear either natively or used as link-layer technology in MPLS networks, which may benefit from CVC to offer multicast support at the link-layer. As it is offered at low levels, the forwarding may be more efficiently implemented, thus accomplishing one of the original goals of MPLS.

8.2 Future work

There are various ways in which this work might be continued. Maybe the most significant step to complete the CVC proposal would be the study of the control part of the mechanism. This topic includes the design and testing of the signaling protocols required to establish CVC-multicast label switched paths. In the general case, the signaling process may be optimized to select the optimum number of PDU multiplexing identifiers to be assigned to each link, which may be different in both directions. One reasonable option is the adaptation of the MPLS Label Distribution Protocol so that it supports our proposal. This broad research issue would deal with the establishment of the multicast tree, membership maintenance, error recovery or reliability, or other similar points introduced in section 2.1.2, where the main requirements of a multicast protocol were discussed. Besides, the internal communication within the LSR between the LDP module and the switching hardware may also studied to complete the proposal.

A more detailed study of the parameters related with the delay, like CDV, packet delay, and packet delay variation would also help in determining in more detail the exact behavior of the CVC mechanism when compared to others, and especially store-and-forward (or VC merge). Related to it, the study of the required buffering in both cases would also help in clarifying the benefits and drawbacks of each of the mechanisms. This study is particularly complex when end-to-end delay parameters are studied in scenarios with many merging stages.

The analysis of more complex and heterogeneous scenarios, and thus, closer to the real ones in diverse environments (local, access or wide area networks) would also be interesting to determine to what extent our proposal and others found in the literature fit in each of these contexts. In this respect, it may be of interest to study the potential application of CVC to emulate broadcast media behavior in the local area or its capacity to aggregate traffic in access networks.

Other interoperability aspects may be also considered so as to allow islands of current equipment to fully interoperate with MPLS ATM LSRs implementing CVC. Some hints were given throughout this work, but more potential scenarios could be studied in more

detail. This would also help in determining what would be the appropriate points of the network to install such native multicasting mechanisms.

Also in the area of MPLS research, the study of the potential links between the traffic engineering work in multicast environments with the operation of the CVC proposal might also shed some light in how it fits in the diverse environments under consideration.

References

[Acharya 97] Acharya A, Dighe R, and Ansari F. 'IP Switching over Fast Atm Cell Transport (IPSOFACTO): Switching Multicast Flows.' Proceedings of Globecom'97: 1850-1854, November 1997.

[Alles 95] Alles A, ATM internetworking, Engineering InterOp, Las Vegas, March 1995.

[Alwayn 01] Alwayn V. 'Advanced MPLS design and implementation.' Cisco Press, September 2001.

[Andrikopoulos 01] Andrikopoulos I, Pavlou G, Georgatsos P, et al. 'Experiments and Enhancements for IP and ATM Integration: The IthACI Project.' IEEE Communications Magazine 39(5): 146-155, May 2001.

[Armitage 96] Armitage G. 'Support for Multicast over UNI 3.0/3.1 based ATM Networks.' IETF RFC 2022. November 1996.

[Armitage 97] Armitage GJ. 'IP multicasting over ATM Networks.' IEEE Journal on Selected Areas in Communications 15(3): 445-457, April 1997.

[Armitage 00] Armitage GJ. 'MPLS: The Magic behind the Myths.' IEEE Communications Magazine 38(1):124-131, January 2000.

[AIC 01] ATM Forum. 'ATM-MPLS Network Interworking. Version 1.0.' Document AF-AIC-0178.000, August 2001.

[IPPNI 96] ATM Forum 'Integrated PNNI (I-PNNI) v1.0 Specification.' Document btd-pnni-ipnni-01.00, December 1996.

[LNNI 99] ATM Forum. 'LAN Emulation over ATM Version 2 – LNNI Specification.' Document AF-LANE-0112.000, February 1999.

[LUNI 97] ATM Forum. 'LAN Emulation over ATM Version 2 – LUNI Specification.' Document AF-LANE-0084.000, July 1997.

[PNNI 02] ATM Forum. 'Private Network-Network Interface Specification. Version 1.1.' Document AF-PNNI-0055.002, April 2002.

[PAR 99] ATM Forum. 'PNNI Augmented Routing (PAR). Version 1.0.' Document AF-RA-0104.000, January 1999.

[Baldi 98] Baldi M, Bergamasco D, Gai S, and Malagrinò D. 'A Comparison of ATM Stream Merging Techniques.' Proceedings of IFIP High Performance Networking (HPN'98): 212-227, Vienna, September 1998.

[Banerjee 02] Banerjee S and Bhattacharjee B. 'A comparison study of application layer multicast protocols.' (Work under submission). Available at: http://www.cs.umd.edu/users/suman/pub-detailed.html, 2002.

[Bolla 96] Bolla R, Davoli F, and Marchese M. 'Evaluation of a Cell Loss Rate Computation Method in ATM Multiplexers with Multiple Bursty Sources and Different Traffic Classes.' Proceedings of IEEE Globecom, pp. 437-441, 1996.

[Boustead 98] Boustead P, Chicharo J, and Anido G. 'Scalability and performance of label switching networks.' Proceedings of GLOBECOM'98, pp. 3029-3034, 1998.

[Braudes 93] Braudes R and Zabele S. 'Requirements for Multicast Protocols.' IETF RFC 1458. May 1993.

[Calvignac 97] Calvignac J, Droz P, Baso C, and Dykeman D. 'Dynamic IDentifier Assignment (DIDA) for Merged ATM Connections.' ATM Forum/97-0504, July 1997.

[Chen 95] Chen TM and Liu SS. 'ATM switching systems.' Artech House Publishers, 1995.

[Chow 99] Chow HK and Leon-Garcia A. 'VC-Merge Capable Scheduler Design.' Proceedings of IEEE ATM Workshop'99: 153-160, 1999.

[Cisco 01] Cisco System, Inc. 'MPLS over ATM: VC Merge.' Available at: http://www.cisco.com/warp/public/121/mpls_vcmerge.html.

[Davie 98] Davie B, Doolan R, and Rekhter Y. 'Switching in IP Networks: IP Switching, Tag Switching, and Related Technologies.' Morgan Kaufmann Publishers, Inc, San Francisco, CA, USA, 1998.

[Davie 01] Davie B, Lawrence J, McCloghrie K et al. 'MPLS using LDP and ATM VC Switching.' IETF RFC 3035, January 2001.

[Diot 97] Diot C, Dabbous W, and Crowcroft J. 'Multipoint Communication: A Survey of Protocols, Functions and Mechanisms.' IEEE JSAC: 277-290, April 1997.

[Deering 89] Deering S. 'Host extensions to IP multicasting.' IETF Request for Comments 1112. August 1989.

[Dumortier 98] Dumortier P, Ooms D, Livens W et al. 'IP Multicast Shortcut over ATM: A Winner Combination.'Proceedings of Globecom'98: 652-657, November 1998.

[Fahmy 97] Fahmy S, Jain R, Kalyanaraman S, et al. 'A Survey of Protocols and Open Issues in ATM Multipoint Communications.' http://www.cis.ohio-state.edu/~jain/papers/mcast.htm, August 97.

[Gauthier 97] Gauthier E, Le Boudec J-Y, and Oeschlin P. 'SMART: A Many-to-Many Multicast Protocol for ATM.' IEEE Journal on Selected Areas in Communications 15(3): 458-472, April 1997.

[Grossglauser 97] Grossglauser M and Ramakrishnan KK. 'SEAM: Scalable and Efficient ATM Multicast.' Proc. of IEEE Infocom'97: 867-875, Kobe (Japan), April 1997.

[Grossman 99] Grossman D. and Heinanen J. 'Multiprotocol Encapsulation over ATM Adaptation Layer 5.' IETF RFC 2684, September 1999. (Obsoletes RFC1483)

[Guo 98] Guo M-H and Chang R-S. 'Multicast ATM Switches: Survey and Performance Evaluation.' ACM SIGCOMM Computer Communication Review 28(2): 98-131, April 1998.

[ITU-T I362] ITU-T. 'B-ISDN ATM Adaptation Layer (AAL) Functional Description.' ITU-T Recommendation I.362. March 1993.

[ITU-T I363.5] ITU-T. 'B-ISDN ATM Adaptation Layer Specification: Type 5 AAL.' ITU-T Recommendation I.363.5. August 1996.

[ITU-T Y1310] ITU-T. 'Transport of IP over ATM in public networks.' ITU-T Recommendation Y.1310. March 2000.

[Kang 00] Kang S, Choi B-C, Choi C-S, Jeong Y-K, and Lee Y-K. 'IP forwarding engine with VC merging in an ATM-based MPLS system.' Proc of 9th International Conference on Computer Communications and Networks, 2000.

[Katsube 97] Katsube Y, Nagami K, and Esaki H. 'Toshiba's Router Architecture Extensions for ATM : Overview.' IETF RFC 2098, February 1997.

[Komandur 97] Komandur S and Mossé D. 'SPAM: A Data Forwarding Model for Multipoint-to-Multipoint Connection Support in ATM Networks.' Proc. of the 6th International Conference on Computer Communications and Networks (IC3N). Las Vegas, September 1997.

[Komandur 98] Komandur S, Crowcroft J, and Mossé D. 'CRAM: Cell Re-labeling At Merge Points for ATM Multicast.' Proc. of IEEE International Conference on ATM (ICATM'98), Colmar (France), June 1998.

[Laubach 98] M. Laubach, J. Halpern. 'Classical IP and ARP over ATM.' IETF RFC2225 (Obsoletes RFC1577), April 1998.

[Lee 99] Lee HH, Kim BI, Lee JS, and Yim CH. 'Structures of an ATM switching system with MPLS functionality.' Proc of Global Telecommunications Conference (Globecom'99), 1999.

[Lee 01] Lee D-W, Lee T-W, Kim Y-C, Choi D-J, and Lee MM-O. 'Implementatiokn of a VC-merge capable crossbar switch on MPLS over ATM.' Proc of Joint 4th International Conference on ATM (ICATM'01) and High Speed Intelligent Internet Symposium, 2001.

[Maher 97] Maher MP and Bhogavilli SK. 'Implementation and Analysis of IP Multicast over ATM.' Proceedings of IEEE Infocom'96: 858-866, Kobe (Japan), April 1997.

[Mangues 99] Mangues-Bafalluy J and Domingo-Pascual J. 'Compound VC Mechanism for Native Multicast in ATM Networks.' Proceedings of the 2nd International Conference on ATM (ICATM'99), pp. 115-124. Colmar (France), June 1999.

[Mangues 00a] Mangues-Bafalluy J and Domingo-Pascual J. 'Analysis of the Requirements for ATM Multicasting based on per PDU-ID Assignment,' Proceedings of IFIP-TC6 Networking 2000. Paris (France), May 2000.

[Mangues 00b] Mangues-Bafalluy J and Domingo-Pascual J. 'Performance Issues of ATM Multicasting based on Per-PDU ID Assignment,' Proceedings of IEEE International Conference on Communications (ICC'00). New Orleans (USA), June 2000.

[Mangues 00c] Mangues-Bafalluy, J. and Domingo-Pascual, J. 'Multicast forwarding over ATM: Native approaches.' IEEE Communications Surveys, 3rd quarter 2000: 2-11, September 2000.

[Mangues 02] Mangues-Bafalluy J and Domingo-Pascual J. 'The Compound VC switch. A non-VC merge ATM multicast switch.' Proceedings of IEEE International Conference on Communications (ICC'02). New York (USA), May 2002.

[Metz 98] Metz C. 'IP routers: New tool for gigabit networking.' IEEE Internet Computing, November, 1998.

[Newman 97] Newman P, Minshall G, Lyon TL, and Huston L. 'IP switching and Gigabit Routers.' IEEE Communications Magazine 35(1): 64-69, January 1997.

[Newman 98] Newman P, Minshall G, and Lyon TL. 'IP switching-ATM under IP.' IEEE/ACM Transactions on Networking 6(2): 117–129, April 1998.

[Ooms 00] Ooms D and Livens W. 'IP Multicast in MPLS Networks.' Proceedings of IEEE Conference on High Speed Performance Switching and Routing' 2000. ATM 2000.

[Ooms 02] Ooms D, Sales B, Livens W et al. 'Framework for IP Multicast in MPLS.' IETF draft: draft-ietf-mpls-multicast-08, April 2002.

[Patzer 00] Patzer J. 'ATM or MPLS – a walk between the hills?.' Presentation from Marconi at GMD Fokus ATM Stammtisch. May 2000. Available at: http://www.fokus.gmd.de/research/cc/tip/atm-stammtisch/protokolle/110500/ATMSTAMM/

[Peterson 00] Peterson LL and Davie BS. 'Computer Networks. A Systems Approach.' 2nd ed. Morgan Kaufmann Publishers, San Francisco, CA, USA, 2000.

[Rosen 01] Rosen EC, Viswanathan A, and Callon R. 'Multiprotocol label switching architecture.' IETF RFC 3031. January 2001.

[Stolyar 99] Stolyar AL and Ramkrishnan KK. 'The Stability of a Flow Merge Point with Non-Interleaving Cut-Through Scheduling Disciplines.' Proceedings of INFOCOM'99, pp. 1231-1238, 1999.

[Stordahl 02] Stordahl K, Kalhagen KO, and Olsen BT. 'Broadband technology demand in Europe.' Proceedings of International Communications Conference for Marketing, Forecasting, and Demand Analysis (ICFC) 2002, June 2002

[Talpade 97] Talpade RR and Ammar MH. 'Multicast Server Architectures for Supporting IP Multicast over ATM.' Proceedings of IFIP Conference on High Performance Networking (HPN'97), April 1997.

[Turner 97] Turner J. 'Extending ATM Networks for Efficient Reliable Multicast.' Proc. of Workshop on Communication and Architectural Support for Network-Based Parallel Computing, Springer Verlag, February 1997.

[Venkateswaran 97] Venkateswaran R, Raghavendra CS, Chen X, and Kumar VP. 'Support for Multiway Communications in ATM Networks.' ATM Forum/97-0316, April 1997.

[Venkateswaran 98] Venkateswaran R, Li S, Chen X, Raghavendra CS, and Ansari N. 'Improved VC-Merging for Multiway Communications in ATM Networks.' Proceedings of the 7th Intl. Conf. on Computer Communications and Networks (IC3N'98), pp. 4-11, 1998.

[Widjaja 98] Widjaja I and Elwalid AI. 'Performance Issues in VC-Merge Capable Switches for IP over ATM Networks.' Proceedings of INFOCOM'98. San Francisco (USA), pp. 372-380, March 1998.

[Widjaja 99] Widjaja I and Elwalid AI. 'Performance Issues in VC-Merge Capable Switches for Multiprotocol Label Switching.' IEEE Journal on Selected Areas in Communications 17(6): pp. 1178-1189, June 1999.

[You 00] You J, Kang S-M, and Chun W. 'Design of the packet forwarding architecture of the ATM based MPLS edge node.' Proc. of IEEE International Conference on Networks (ICON'2000), 2000.

[Zhou 99] Zhou P and Yang OWW. 'Reducing buffer requirement for VC-merge capable ATM switches.' Proceedings of GLOBECOM'99, pp. 44-48, 1999.

Bibliography

Other previous work which served as reference during the realization of this thesis but does not appear referrenced along the text follows:

[Bertsekas 97] Bertsekas D and Gallager R. 'Data Networks.' Prentice-Hall International Editions, 1987.

[UNI 96] ATM Forum. 'ATM User-Network Interface (UNI) Signalling Specification. Version 4.0.' Document AF-SIG-0061.000, July 1996.

[MPOA 97] ATM Forum. 'Multi-Protocol over ATM. Version 1.0'. Document AF-MPOA-0087.000. July 1997.

[Comer 95] Comer DE. 'Internetworking with TCP/IP. Volume I: Principles, Protocols, and Architecture.' 3rd ed. Prentice-Hall International, Englewood Cliffs, CA, USA, 1995.

[Dumortier 98] Dumortier P. 'Toward a New IP over ATM Routing Paradigm.' IEEE Communications Magazine 36(1): 82-86, January 1998.

[Finn 96] Finn N., and Mason T. 'ATM LAN Emulation.' IEEE Communications Magazine. Junio 1996.

[Jain 97] Jain R. 'Multipoint Communication Over IP and ATM.' Audio Recordings of Professor Jain's Lectures. Available at: http://www.netlab.ohio-state.edu/cis788-97/h_amptv.html

[Keshav 98] Keshav S. and Sharma R. 'Issues and Trends in Router Design.' IEEE Communications Magazine 36(5): 144-151, May 1998.

[Lea 99] Lea C.-T., Tsui C.-Y., Li B, et al. 'A/I Net: A network that integrates ATM and IP.' IEEE Network 13(1): 48-55, January/February, 1999.

[Pung 98] Pung HK and Leow AS. 'A Multipoint-to-Multipoint Multicast over ATM supporting Heterogeneous QoS.' Proceedings of the 7th Intl. Conference on Computer Communications and Networks (IC3N), pp. 609-613, 1998.

[Schmid 98] Schmid AL, Iliadis I, and Droz P. 'Impact of VC Merging on Buffer Requirements in ATM Networks.' Proceedings of IFIP High Performance Networking (HPN'98): 212-227, Vienna, September 1998.

[Stallings 97] Stallings W. 'Comunicaciones y Redes de Computadores.' 5th ed. Prentice Hall. Madrid, 1997.

[Tanenbaum 96] Tanenbaum AS. 'Computer Networks.' 3rd ed., Prentice Hall, 1996.

[Walrand 00] Walrand J and Varaiya P. 'High-Performance Communications Networks. Chapter 12: Switching.' 2nd ed. Morgan Kaufmann Publishers, San Francisco, CA, USA, 2000.