NEAR EAST UNIVRSITY

Faculty of Engineering

Department of Computer Engineering

WIRELESS ASYNCHRONOUS TRANSFER MODE

Graduation Project
Com400

Student: Mohamad Shehadi AL-SHARAF

Supervisor: Asst. Prof. Dr Firudin Muradov

NICOSIA 2005

ACKNOWLEDGMENT

At the beginning all my thanks to my **'Great God'**. I wish to sincerely thank all of those who helped me to do my project. Of course I'm thanking my project's advisor Asst. Prof. Dr. Firudin Muradov under his guidance where I got my quick progress from his advices. My thanks for him not just of helping me in doing my project, but because he always helps me since 4 yeas I spent in Near East University.

Moreover I'll not forget my friends who were helping me to do this project: Amir Sarayreh, Noor Zuaiter, Firas Bhar, Mahmoud Ghuneim, Khaled Ghuneim, Rami Amar, Sa'ad AL Mohiesen, Mohamad Sarhan, Youif Abadi, Yousif Ghuneim, and Mohamad Majzob , Nagham Badawi and Yousif Abu Khurj.

And also I'll never forget the person who was encouraged and always support me to study hard to be a successful man, for Mrs. [redacted] Aid.

Finally for many others who supported me during my education years where the words are not enough to thanks them or to describe how much I love them and doing my best for them "My Father and My Mother". Also my Brothers they are always beside me in any time, Thanks lot dears.

ABSTRACT

For a large number of applications, there is a strong need for the provision of a wireless service by the B-ISDN. However, the cell-based ATM, which is used for switching and multiplexing in this public telecommunication network, is a connection-oriented technique. The B-ISDN should therefore be extended with functionality to allow for wireless communications. This dissertation addresses the design and analysis of such an extension.

An architectural framework is presented, which places the protocols to be used in perspective. Two possible network architectures result from the functional decomposition of the wireless service into cooperating protocol entities and the underlying ATM service. In the first one, end-systems of the B-ISDN are interconnected by means of end-to-end ATM connections. In the second one, end systems are connected to special entities in the B-ISDN, called Connectionless Servers (CLSs). The CLSs are interconnected by ATM connections, thus constituting a wireless overlay network on top of ATM.

A number of different implementation architectures for a CLS are proposed, and analyzed with respect to effectiveness, availability, scalability, and in particular, performance. The major distinction between these implementation architectures is the distribution of functionality over modules. Furthermore, two different modes of operation are identified for a CLS. In the message mode of operation, a packet is reassembled from the incoming cells before it is processed and forwarded. In streaming mode of operation, the first cell of a packet is immediately processed and forwarded upon arrival, while state information is maintained for the processing and forwarding of subsequent cells of the packet.

A number of performance models are developed in this dissertation. An approximate model of a CLS is analyzed to allow for comparison of the delay which is experienced by cells for different implementation architectures and modes of operation. If the bandwidth

assigned to ATM connections between CLSs is relatively high, message mode of operation yields the lowest delay, otherwise streaming mode performs best.

In order to support the dimensioning of a reassembly buffer in a CLS operating in message mode, another, more detailed model is developed and analyzed. It allows the computation of the packet loss probability of a buffer, as a function of its size.

An essential function for the provision of a wireless service using ATM is connection management. This function instructs the signalling system of the B-ISDN to establish and release ATM connections as needed for the transfer of packets. A new mechanism is proposed that exploits the expected correlation between subsequent packet arrivals to reduce the average bandwidth that needs to be reserved by the ATM network.

A performance model is developed and analysed to determine the optimal control parameters of the new mechanism, and to evaluate its behaviour. It is shown that bandwidth reductions of up to 95% can be obtained, compared to conventional mechanisms, without affecting the average delay experienced by packets.

LIST OF ABBREVIATIONS

A

AAL - ATM Adaptation Layer
AL - Alignment
ATM - Asynchronous Transfer Mode
AUU - ATM-user-to-ATM-user Indication

B

BASize - Buffer Allocation Size
BASTA - Bernoulli arrivals see time averages
BCDS - Broadband Connectionless Data Service
BCL - Broadband Connectionless
BCLS - Broadband Connectionless Service
B-ISDN - Broadband Integrated Services Digital Network
BOM - Beginning of Message
BTag - Begin Tag

C

CAC - Connection Admission Control
CATV - Cable Antenna Television
CBDS - Connectionless Broadband Data Service
CEP - Connection Endpoint
CIB - CRC Indication Bit
CLNAP - Connectionless Network Access Protocol
CLNIP - Connectionless Network Interface Protocol
CLNP - Connectionless Network Protocol
CLP - Cell Loss Priority
CLS - Connectionless Server
CLSM - Connectionless Service Module
CO - Connection-oriented
COM - Continuation of Message
CPCS - Common Part of the Convergence Sublayer
CPI - Common Part Indicator
CpP - Connection per Packet
CRC - Cyclic Redundancy Check
CS - Convergence Sublayer
CTMC - Continuous-Time Markov Chain
CUG - Closed User Group

D

D-MAP - Discrete-time Markovian Arrival Process
DTMC - Discrete-Time Markov Chain
DQDB - Distributed Queue Dual Bus

E

EOM - End of Message
ETag - End Tag
ETSI - European Telecommunications Standards Institute

F

FIFO - First In First Out

G

GFC - Generic Flow Control

H

HDTV - High Definition Television
HEC - Header Error Control
HEL - Header Extension Length
HLPI - Higher Layer Protocol Indicator

I

ICIP - Inter-exchange Carrier Interface Protocol
IDU - Interface Data Unit
i.i.d - independent identically distributed
IP - Internet Protocol
IPP - Interrupted Poisson Process
IS - Interconnection Structure
ISO - International Organization for Standardization
ITU - International Telecommunication Union
IWU - Interworking Unit

L

LAN - Local Area Network
LI - Length Indication
LOTOS - Language Of Temporal Ordering Specification

M

MAC - Medium Access Control
MAN - Metropolitan Area Network
Mbits - Megabits
MID - Multiplexing Identification
MMPP - Markov Modulated Poisson Process

N

NNI - Network Node Interface
NAP - Network Access Protocol
NP - Network Protocol

O

OCDR - On-demand Connection with Delayed Release
OSI-RM - Reference Model for Open Systems Interconnection

P

PAD - Padding
PASTA - Poisson arrivals see time averages
PC - Permanent Connection
PCI - Protocol Control Information
PDU - Protocol Data Unit
PL - Physical Layer
PPM - Packet Processing Module
PRM - Protocol Reference Model

PT - Payload Type

Q

QNA - Queueing Network Analyzer
QoS - Quality of Service

R

RACE - Research and development in Advanced Communications technologies for Europe
RM - Routing Module
RPC - Remote Procedure Call

S

s - second
SAP - Service Access Point
SAR - Segmentation and Reassembly (sublayer)
SDU - Service Data Unit
SIP - SMDS Interface Protocol
SMDS - Switched Multi-megabit Data Service
SN - Sequence Number
SSCS - Service Specific Part of the Convergence Sublayer
SSM - Single Segment Message
ST - Sequence Type

T

TCP - Transport Control Protocol
TV - Television

U

UNI - User-Network Interface
UPC - Usage Parameter Control
UU - User-to-User indication

V

VBC - Variable Bandwidth Connection
VC - Virtual Channel
VP - Virtual Path
VCC - Virtual Channel Connection
VCI - Virtual Channel Identifier
VCL - Virtual Channel Link
VPC - Virtual Path Connection
VPI - Virtual Path Identifier
VPL - Virtual Path Link
VPN - Virtual Private Network

W

WAN - Wide Area Network

TABLE OF CONTENTS

INTRODUCTION

Asynchronous Transfer Mode (**ATM**) has been accepted universally as the transfer mode of choice for Broadband Integrated Services Digital Networks (**BISDN**). ATM can handle any kind of information i.e. voice, data, image, text and video in an integrated manner. ATM provides good bandwidth flexibility and can be used efficiently from desktop computers to local area and wide area networks. ATM has been advocated as an important technology for the wide area interconnection of heterogeneous networks. In ATM networks, the data is divided into small, fixed length units called cells. The cell is 53 bytes. Each cell contains a 5 byte header; this header contains the identification, control priority, and routing information. The other 48 bytes are the actual data. ATM does not provide any error detection operations on the user payload, inside the cell, and also offers no retransmission services. ATM is a connection- oriented packet switching technique in which all packets are of fixed length i.e. 53 bytes(5 bytes for header and 48 bytes for information). No processing like error control is done on the information field of ATM cells inside the network and it is carried transparently in the network.

ATM meets the following objectives for BISDN networks.

1) Supports all existing services as well as emerging services in the future

2) Utilizes network resources very efficiently

3) Minimizes the switching complexity

4) Minimizes the processing time at the intermediate nodes and supports very high transmission speeds.

5) Minimizes the number of buffers required at the intermediate nodes to bound the delay and the complexity of buffer management

6) Guarantees performance requirements of existing and emerging applications

The First Chapter introduces concepts that are essential to the dissertation, such as ATM and B-ISDN, and expands on the notion of a connectionless service. Furthermore, it explores the applications the connectionless service is being designed for, and surveys related work in the literature.

While the chapters 2, and 3, describes the design and functional analysis of the connectionless service.

Chapter 2 presents the design of the network architecture, starting from the requirements.

Chapter 3 is concerned with the design of a CLS. It presents several alternative implementation architectures, and analyses them with respect to their effectiveness, availability, and scalability.

The last part of this Project are summarizes the results of the work, and draws conclusions.

CHAPTER ONE

CONCEPTS, APPLICATIONS, AND RELATED WORK

A lot of effort is currently being invested in the design of broadband communication systems. The design of a system for supporting wireless communications should be seen as an integral part of these activities. In this chapter we review some basic concepts in B-ISDN, discuss applications that will use a wireless service, and survey related work. The purpose of this chapter is to introduce the information that forms a starting point for the work presented in the rest of the Project.

The organization of this chapter is as follows. First, in Section 1.1, a number of concepts that are relevant to the Project are introduced. In Section 1.2 we look at applications that may want to use wireless communication. In Section 1.3 we survey related work. We conclude this chapter by summarizing it, and giving concluding remarks, in Section 1.4.

1.1Basic Concepts

The concept of wireless communication is essential to this work, and can only be understood in relation to its opposite, i.e., connection-oriented communication.
Both are discussed in Section 1.1.1. The network we are considering for providing wireless data communication is the B-ISDN, which will be introduced in Section 1.1.2. Finally, in Section 1.1.3, we will briefly review some basic concepts of ATM, which is the network technology that will be used to support wireless data communication.

1.1.1 Wireless versus Connection-oriented

In this Project the terms wireless and connection-oriented are used as a characteristic of a service, i.e., the observable behaviors of a communication system. A service is wireless if two or more users of the service can transfer data using the communication system without first establishing and later releasing a connection. A service is connection-oriented if users must establish a

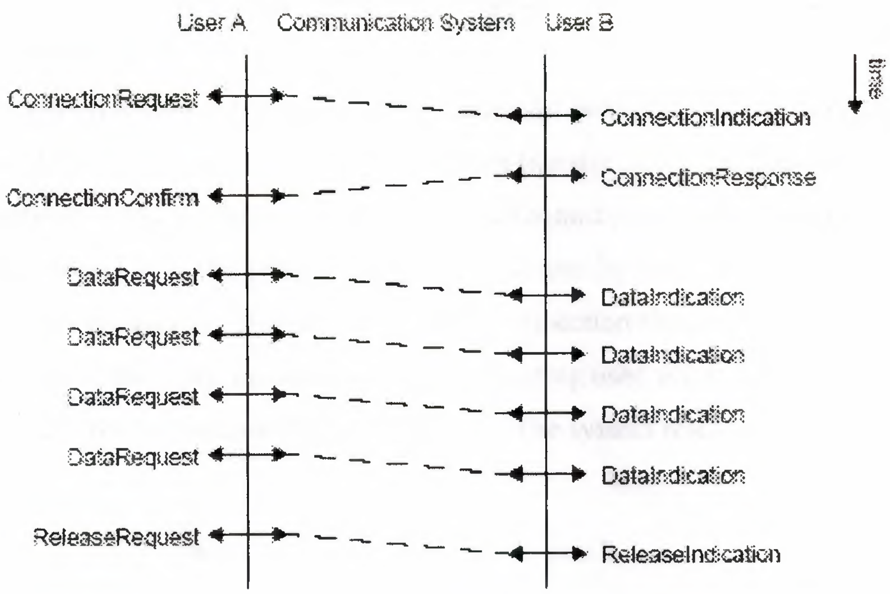


Figure 1-1: Sequence of Service Primitives for a Connection-oriented Service

connection before they can transfer data using the communication system. The establishment of a connection is a negotiation between the users who wish to communicate and the communication system. During the negotiation, state information related to the connection is exchanged between the parties. The communication system reserves resources for a connection, e.g., bandwidth.

The essential interactions between a communication system and its users can be described by means of the sequence of service primitives, which are exchanged at the boundaries between the system and its users. Figure 1-1 shows a likely sequence of service primitives in case of a connection-oriented service, where two users want to communicate. Service primitives are denoted by double arrows, to indicate an interaction in which both the user and the communication system are involved. First, the initiating users request the system to establish a connection (ConnectionRequest). In this request at least the address of the required destination and often parameters regarding the characteristics of the traffic and the required Quality of Service (QoS) are passed to the system. The system analyses the parameters of the request and determines to what extent it can support the requested connection. If the connection can be supported, a ConnectionIndication primitive occurs between the system and the called user. This user checks whether and under what conditions it wants to accept the connection. The acceptance of the connection is indicated to the system by another interaction (ConnectionResponse).

Finally the initiating user is informed of the successful establishment of the connection (ConnectionConfirm). From now on, the users can transfer data. The data is passed from the sending user to the system by means of a DataRequest primitive, transported by the system, and passed from the system to the receiving user by means of a DataIndication primitive. After all data has been transferred, the connection should be closed again. Therefore, one of the users, not necessarily the initiating user, informs the system that it wants to release the connection (ReleaseRequest). The system releases the connection

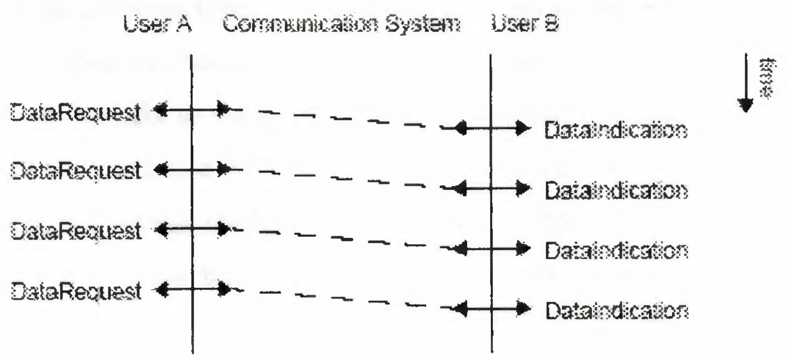


Figure 1-2: Sequence of Service Primitives for a wireless Service

and informs the other user (ReleaseIndication).

For a wireless service, the sequence of service primitives is much simpler (Figure 1-2). No primitives for establishing or releasing a connection are needed. Each data unit is transferred individually. The sending user passes the data to the system by means of a DataRequest primitive. The system transports the data and passes it to the receiving user by means of a DataIndication primitive. All information regarding the transfer of the data unit, e.g., required destination address, should be passed between the users and the network in these primitives.

An essential characteristic of a wireless service is that the system does not have any advance knowledge about the data that should be transferred (e.g., destination, or rate at which data units are offered to the system). In case of a connection-oriented service this

information is agreed upon during the connection establishment, so that the system can reserve resources for the transfer of the data units.
For both types of service there are classes of applications that they support best. Applications that require a direct association between the users, such as telephony, are best served by a connection-oriented service. Other applications, such as those which involve the transfer of only a single unit of information between a source and a destination are better served by a wireless service.
Moreover, there are a huge number of applications that use the TCP/IP protocol suite. These applications require a wireless service at the network layer, also when they will be supported by the B-ISDN.

It is possible to use different types of services at different layers in the network. In [82] the use of wireless protocols at the network layer, to support all types of applications is advocated. Others prefer to use a single connection-oriented protocol, i.e., ATM, to support all applications. Since ATM has been standardized for future integrated networks, we assume ATM as the basic method to serve all applications. However, we believe that additional provisions have to be made in the network to accommodate applications that are wireless in nature.

1.1.2 B-ISDN

The Broadband Integrated Services Digital Network (B-ISDN) is expected to be the major future telecommunications network for the wide area. It will provide a wider range of possible applications, and support much higher throughputs than present telecommunication networks. Applications, which can be very diverse in nature, will all be supported by a single network. Moreover, a user can access the network via a single standardized interface, the User-Network Interface (UNI).
The applications that should be supported by the B-ISDN are not only very diverse; they also put very diverse requirements on the network. The network should support:

- Point-to-point as well as multi-point communications;
- Single-medium as well as multimedia communications;
- Wireless as well as a connection-oriented communications;

• Narrowband as well as broadband communications;
• Communications involving constant bit rate as well as variable bit rate traffic; and
• Communications for applications with a very diverse range of QoS requirements.

Some examples of applications that are foreseen for the B-ISDN are video-conferencing, high Definition Television (HDTV) distribution, video-on-demand, telephony, and applications that are currently supported by the Internet.

In order to support all these applications, the Asynchronous Transfer Mode (ATM) has been adopted for multiplexing and switching in the network ([11]).
ATM is a technique, where all user information is transferred in Protocol Data Units (PDUs) of fixed size, called cells. Cells are identified by means of a header, so that

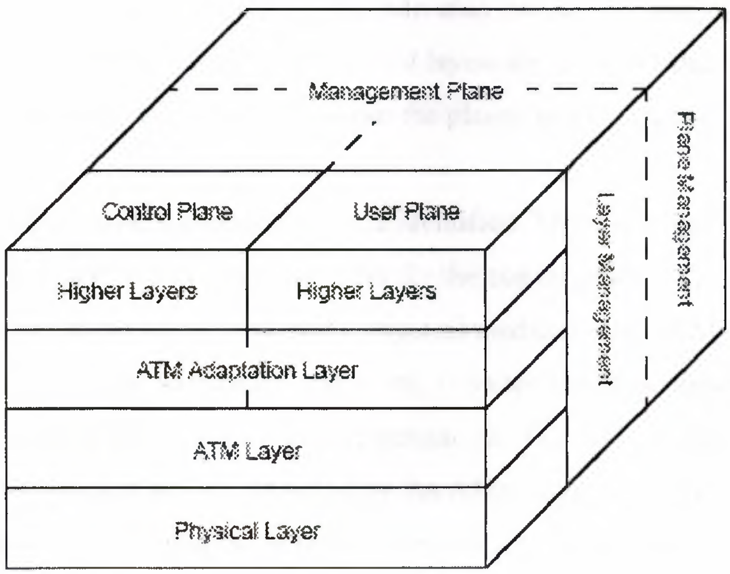


Figure 1-3: The B-ISDN Protocol Reference Model

they can be identified for (de)multiplexing and switching. ATM should also be used at the UNI of the B-ISDN. Two different UNIs have been defined ([82]); one with a bit rate of 155.520 Mbits/s, and one with a bit rate of 622.080 Mbits/s.

A Protocol Reference Model (PRM) has been defined for the B-ISDN in [18] (see Figure 1-3). The PRM is divided into planes and layers. The division into (vertical) planes is done to identify different types of functionality, i.e., for the transfer of user information, for the control of calls and connections, and for management. The division into (horizontal) layers is done to create a stepwise independency between the medium, used for the transmission of signals, and the applications. A layer uses the next lower layer to provide a certain, less medium dependent, and more application dependent, service to the next higher layer.

The B-ISDN PRM distinguishes between three planes. The user plane contains the functions that deal with the user information. The control plane contains functions for the control of calls and connections, and the transport of information on behalf of these functions (signalling). Finally, the management plane provides for the coordination between user and control plane, and for the management of individual entities and the overall network. The management plane has been subdivided into layer management, performing management functions for specific protocol layers and their entities, and plane management, performing coordination between the planes and management of the system as a whole.

Within the user and control plane, protocol layers are identified. The lower layers are common to both planes. The Physical Layer provides for the convergence of ATM cells to signals that can be transferred over a physical medium. The ATM Layer provides for the end-to-end transfer of cells along a connection. It performs switching and multiplexing of cells from different connections. The ATM Adaptation Layer (AAL) adapts the common service provided by the ATM Layer to a service that can better support specific classes of applications. It provides for instance for segmentation and reassembly and for synchronization of source and destination.

Different views exist for relating the layering adopted in the B-ISDN PRM and the layering of the Reference Model for Open Systems Interconnection (OSI-RM) ([68]). In the context of this Project, it is convenient to situate the AAL in the OSI Datalink Layer. The ATM Layer can be considered as an upper sublayer of the OSI Physical Layer, or as a lower sublayer of the Datalink Layer.

The applications to be supported by the B-ISDN have very diverse communication requirements. The ATM Layer is common to all applications. The purpose of the AAL is to adapt the ATM service to the service required for the specific applications requirements. In order to come up with a limited number of AAL services, applications have been classified according to the following communication requirements:

• Timing relation between source and destination (required or not required);
• Bit rate (constant or variable); and
• Connection mode (connection-oriented or wireless).

Since not all combination of the above requirements are foreseen, only four service classes are distinguished according to Table 2-1. Clearly, the class of AAL service we are interested in for the support of wireless data communications over ATM is service class D. This class does not provide a timing relation between source and destination, supports variable bit rate traffic, and is wireless.

Table 2-1: AAL Service Classes

	Class A	Class B	Class C	Class D
Timing relation between source and destination	Required		Not Required	
Bit Rate	Constant	Variable		
Connection Mode	Connection-oriented			Connectionless

For each of the service classes, one or more protocols have been defined that can provide the service. For class D, two protocols, called AAL 3/4 and AAL 5 have been defined ([16], [65]) 1. These protocols will be elaborated on in Section 2.4. For further reading on B-ISDN we refer to [45].

1.1.3 ATM

The Asynchronous Transfer Mode (ATM) is a technique that is used for switching and multiplexing in the B-ISDN. It can route fixed-size data units, called cells, through a network of switches interconnected by links, from source to destination.

ATM is connection-oriented, i.e., prior to the transfer of cells, a connection is established through the network, and cells are forwarded along the connection subsequently. Multiplexing of cells from different connections on an ATM link is done using asynchronous time division. Time is divided into slots, in which a single ATM cell fits. Slots can be assigned to connections asynchronously, i.e., whenever a slot is needed for a connection it can be used. Identification of the connection a cell belongs to is done by means of a label in the header of the cell. Figure 1-4 gives an example of a series of cells, which are transmitted on a link consecutively. All cells with the same label, e.g., 7, are identified as belonging to a certain connection.

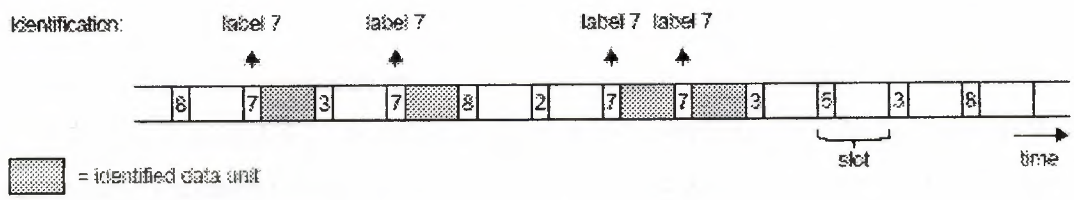


Figure 1-4: Asynchronous Time Division Multiplexing

The alternative to asynchronous time division multiplexing is synchronous time division multiplexing, which is used in present telecommunication networks.
Time is again divided into slots, and slots are grouped into frames. Slots are assigned to a connection synchronously, i.e., a connection is assigned the same number of slots in each frame. Identification of the connection a slot belongs to is done by means of the position of a slot in the frame. Figure 1-5 gives an example of a series of consecutive cells in

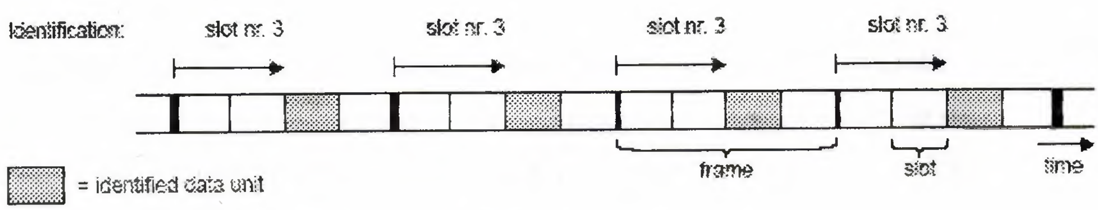


Figure 1-5: Synchronous Time Division Multiplexing

this case. All cells which are the third one of a frame are identified as belonging to a certain connection.

An important advantage of asynchronous over synchronous time division multiplexing is that the transmission capacity assigned to a connection can vary in time, depending on the needs of the application. Furthermore, unlike with synchronous time division multiplexing, where the assigned capacity is an integer multiple of the smallest possible capacity (one slot assigned per frame), any capacity can be assigned to a connection.

ATM switches receive cells on a number of incoming links. The connection an incoming cell belongs to is uniquely defined by the incoming link, and the label carried in the cell header. In the switch a routing table is maintained, in which the outgoing ATM link and the required label for the connection on that ATM link is given for each connection.

Using this table, the switch can replace the label of a cell, and forward it to the proper outgoing link. Figure 1-6 gives an example of ATM switching. The shaded entry in the routing table indicates that all cells that arrive on link 2 with label 7 should be forwarded to link 4, while the label should be modified to 3. ATM cells are drawn as in Figure 1-4, i.e., from left to right, first the header, which contains the label, and then the payload. As stated before, ATM is a connection-oriented technique. The header of an ATM cell relates the cell to a previously established ATM connection. Two types of ATM connections are identified, Virtual Path Connections (VPCs) and Virtual Channel Connections (VCCs). Therefore, also two identifiers can be found in the header of the cell, a Virtual Path Identifier (VPI) and a Virtual Channel Identifier (VCI).

The combination of the two, referred to as VPI/VCI, determines the ATM connection a cell belongs to.
A physical link between two switches carries a number of Virtual Path Links (VPLs), each of them is identified by a VPI. The concatenation of a number of VPLs forms a Virtual Path Connection (VPC). Within a VPC, a number of Virtual Channel Links (VCLs) can be identified. Each VCL is identified by a VCI, which is

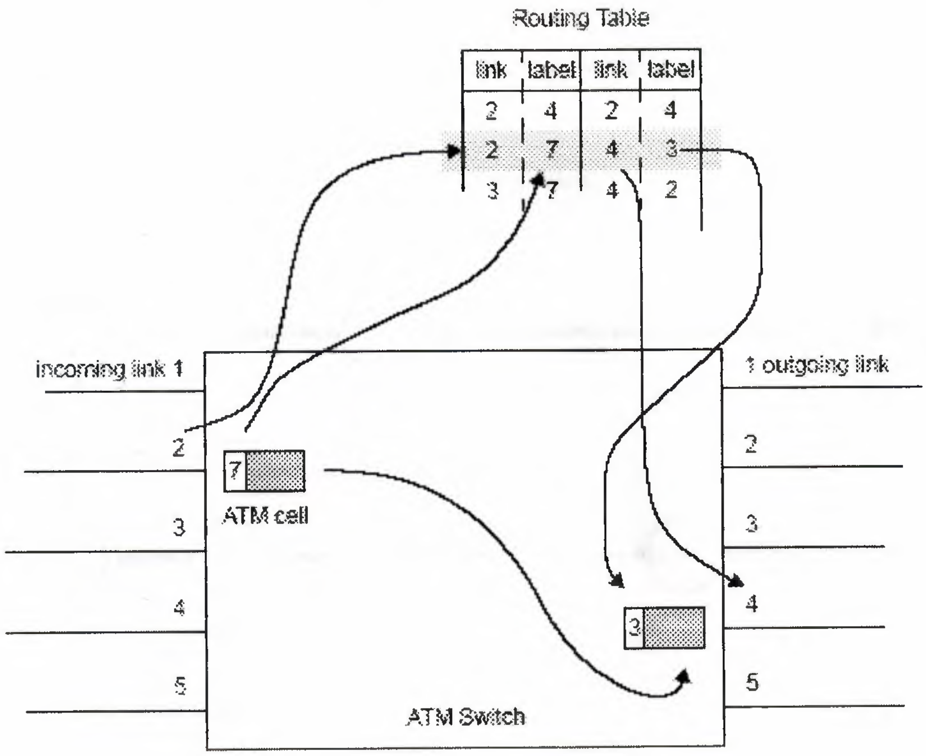


Figure 1-6: ATM Switching

unique for that VPC. The concatenation of a number of VCLs forms a Virtual Channel Connection.

Figure 1-7 shows the relationship between the different types of links and connections. In the figure, a series of 7 ATM switches is shown. On a link between a pair of switches, a number of VP links can be identified. A concatenation of VP links forms a VP connection, e.g., from the first to the third switch. All switches operate on VPs; only some of them operate also on VCs. These are also visible at the VC level. Within a VP connection (between two VC switches), a number of VC links can be identified. The concatenation of a number of these links forms a VC connection, e.g., between the first and the seventh switch.

An ATM cell has a payload field of 48 octets. The header of the cell is 5 octets long. The major fields in the header are the VPI and VCI field. Furthermore, the header contains a Header Error Control (HEC) field to protect the header against bit errors. A Cell Loss Priority (CLP) field can be used to indicate the relative importance of the cell in the connection. The Payload Type (PT) field carries additional information, e.g., an ATM-user-to-ATM-user indication that is transported transparently by the network. At the UNI, also a Generic Flow Control (GFC) field is present in the header in order to control the access to a shared link.

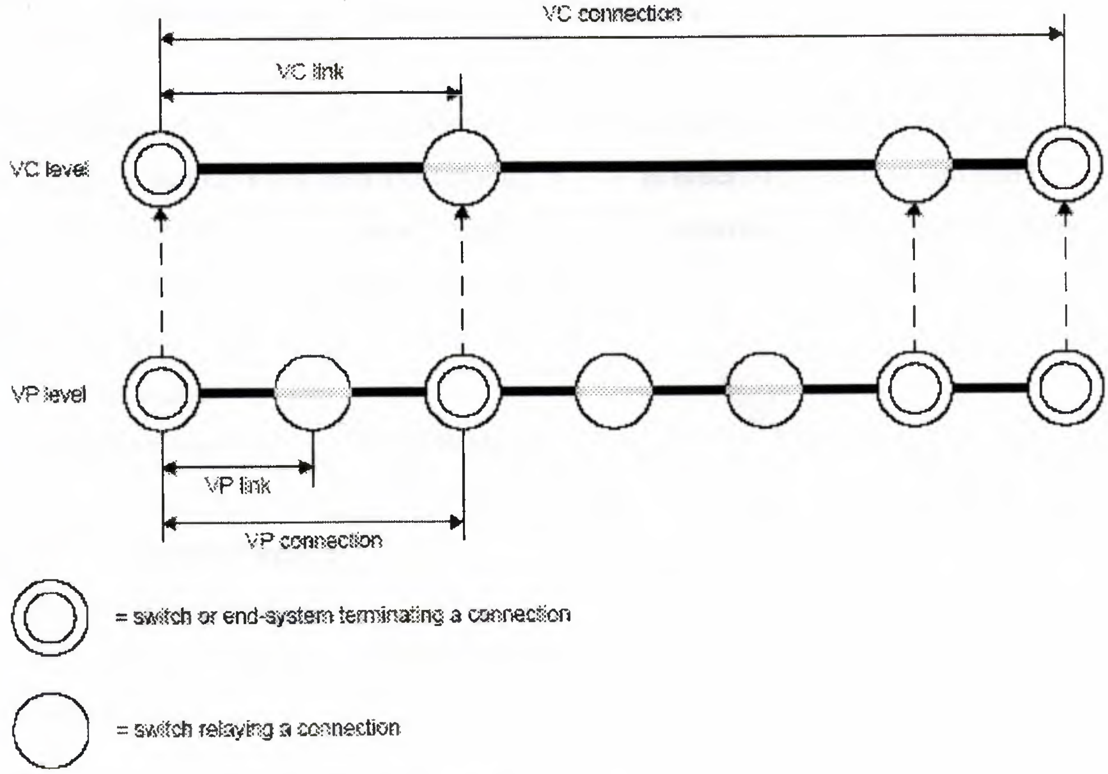


Figure 1-7: ATM Connections

Error control should be performed in layers above ATM on an end-to-end basis. Only errors in the header are detected or corrected (depending on the mode of operation) by means of the 8 bit HEC field, if possible.

In order to protect an ATM network against overload and guarantee the requested QoS to the users, bandwidth is reserved for each connection. A Connection Admission Control (CAC) mechanism checks whether or not bandwidth for a newly requested connection is available. If not, the connection is refused. If the bandwidth is available, the connection is established, and the bandwidth is reserved. In order to check if the user does not violate the agreed bandwidth another mechanism, called Usage Parameter Control (UPC), has to be implemented at the border of (the public part of) the network, directly after the UNI.

Cells that cause a violation of the agreed bandwidth are either discarded directly, or given a low cell loss priority, so that they are the first ones to be discarded in case of congestion.

1.2 Applications using Wireless Communications

The emphasis in this Project is on wireless communications over the wide area, with throughputs that have not been available up to now. In order to provide insight into the requirements on the system, we will explore the characteristics of applications that have been identified in a number of publications, e.g.

- file transfer,
- Terminal access,
- Information retrieval (e.g., World Wide Web),
- Computer graphics,
- distributed supercomputing,
- Remote procedure call (RPC),
- Virtual memory page swapping and paging, and
- Electronic mail.

Although not all applications listed here need necessarily be supported by a wireless network, most of them are wireless in nature. This means that the applications need to transfer one or more individual pieces of information, not a stream of related pieces of information. Electronic mail is an example of an application that is inherently wireless. Its purpose is to transfer individual messages from source to destination. There is no need for an association between source and destination before the message is transferred.
The end-systems that implement the applications can be attached directly to the B-ISDN. However, most applications have been first introduced in a local environment, so that a lot of existing end-systems have been attached to Local Area Networks (LANs). Another reason for attaching end-systems to LANs is the locality of a large share of the generated traffic.
However, there is a growing demand for communications over a wider area. This can be accommodated by interconnecting LANs by means of networks that span a wider

geographical area. Most LANs use wireless protocols. In order to avoid complicated and costly protocol conversion, it is often desirable to interconnect these LANs by means of wireless networks. A first stage into this development is

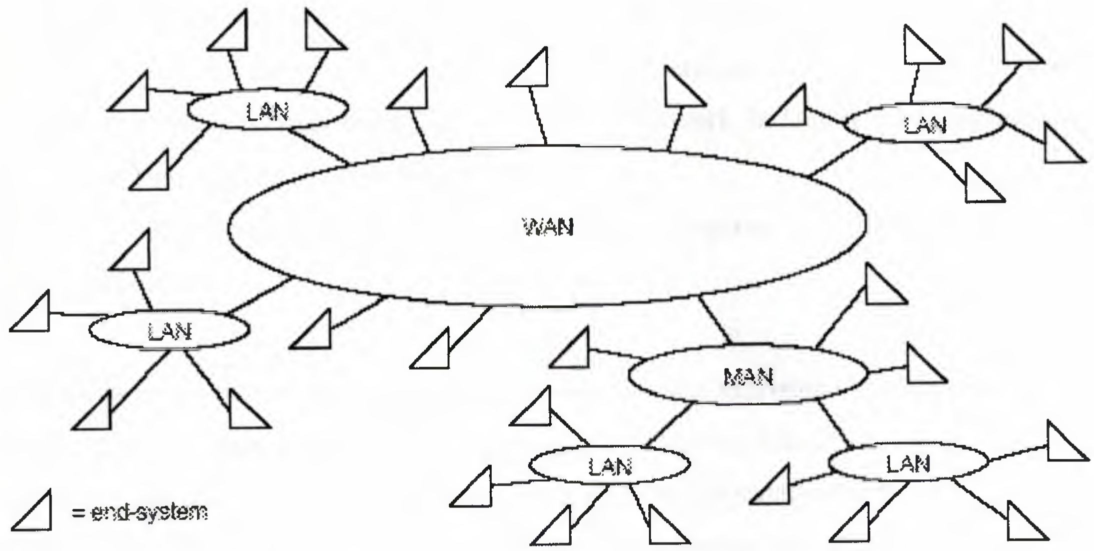


Figure 1-8: Connection of End-systems, LANs, and MANs to a WAN

the interconnection of LANs by Metropolitan Area Networks (MANs), which typically have a geographic span of about 50 kilometres. Many of these MANs are based on the Distributed Queue Dual Bus (DQDB) technique, defined in the IEEE 802.6 standard ([65]).

If a Wide Area Network (WAN) supporting broadband wireless communications becomes available, it can be used to interconnect these MANs, and to interconnect LANs directly. Furthermore, end-systems with high communication needs can be attached directly to the WAN. This will lead to a scenario where end-systems are attached to either a LAN, a MAN, or a WAN, with LANs being connected to either a MAN or a WAN, and the MANs connected to the WAN (see Figure 1-8).

We refer to the systems that interconnect different networks, and perform the needed conversion, as Interworking Units (IWUs). These IWUs can function as routers, interconnecting the different networks at the network layer, e.g., using the Internet Protocol (IP) or the ISO Connectionless Network Protocol (CLNP) ([68]). The IWU can also function as a bridge, interconnecting the different networks at the medium access control (MAC) sublayer.

It is expected that most traffic to be transported by the wide area wireless network will initially come from end-systems attached to LANs. Interconnection of these LAN is often referred to as the most important application of the network. In fact, the real applications to be supported are those listed above.

From the point of view of the WAN, the LAN only aggregates the traffic generated by these applications.

In order to get more insight in the traffic generated by the individual applications, we consider the requirements on the transfer of an single unit of information. [37] and [102] list for a number of applications the expected size of an information unit.

Furthermore, they give an indication of the required response time for these applications. Figure 1-9 graphically represents these requirements. For each application an area of requirements is given, expressing both the uncertainty and the variation in requirements.

Note that the information units sizes and response time requirements are given at the application level. An information unit may very well be transported in a number of smaller packets at the network level.

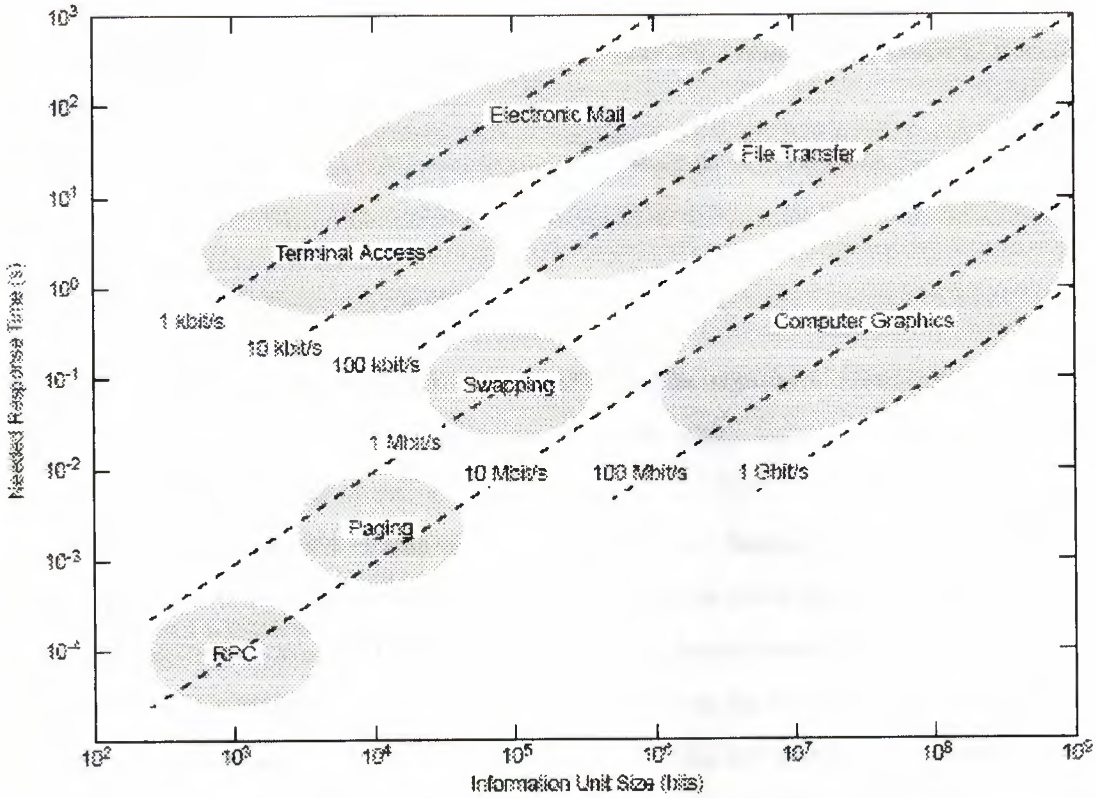


Figure 1-9: Applications Requirements (based on [37] and [102])

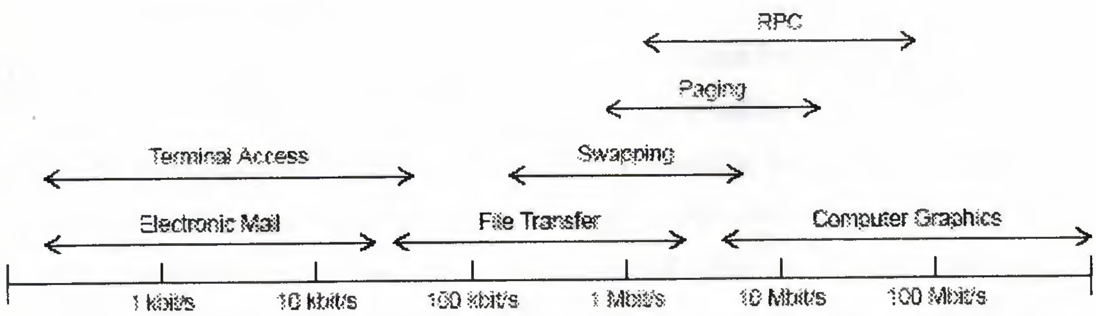


Figure 1-10: Throughput Requirements of Applications (based on [37] and [102])

In Figure 1-9 a number of diagonal lines have been drawn that give an indication of the throughput required transferring information units of a certain size within the response time. Figure 1-10 summarizes these throughput requirements. It can be seen from the figure that throughput requirements range from less than a kilobit per second for electronic mail and terminal access to a gigabit per second for computer graphics.

In principle, the given applications do not tolerate any loss or corruption of data.

This does not imply that no loss or corruption can be tolerated from the wireless service we are designing. End-to-end protocols used on top of the wireless service (e.g., a transport protocol) can enhance the provided QoS.

Practical values for these and QoS requirements will be given in Section 2.1.

1.3 Prior and Related Work

Some of the related work, especially which published by standardization organizations or interest groups, has been a starting-point for our work. Other work has a different scope, or is complementary to our work.

The first work on integrating datagram (i.e., wireless) and connectionoriented (e.g., voice) communications in a single network based on a fast packet switching technique, performed at AT&T Bell Laboratories, has been reported by Turner. At the same time Coudreuse et al. of France Telecom worked on the asynchronous time division multiplexing technique ([19]). Eventually, these activities led to the definition of ATM, first within the Telecommunication standardization section of the International Telecommunication Union (ITU-T, the former CCITT) ([11], [70], [76]), and later on also in the ATM Forum.

ATM has been developed as a connection-oriented technique for, among others, the following reasons. Because of the required high-speed operation of the network, the time needed for processing in the nodes, e.g., for routing should be kept low. Furthermore, many applications require a guaranteed QoS, which can be individually negotiated for a specific application process. After the establishment of ATM, as a basically connection-oriented technique, the problem of supporting wireless communications using ATM became of significant interest. A specification of the service to be provided to the user and the access to it has been given by a number of different organizations. Bellcore has defined the Switched Multi-megabit Data Service (SMDS), which should initially be provided using MANs and later using ATM. The European Telecommunications Standards Institute (ETSI) has defined the european version of SMDS, called Wireless Broadband Data Service (CBDS) in [32], [33], [34], and [35]. Finally, the ITU has

defined the Broadband Wireless Data Service (BCDS) to be provided using an ATM-based B-ISDN ([11], [15], [72]).
Currently, recommendations on how to provide the BCDS are being defined within the ITU ([16], [78], [79]). They mainly focus on the protocols to be used on top of ATM for the provision of the BCDS. Furthermore, architectural issues are considered. A general overview of these issues is given in [22].
A number of articles have been published related to the protocols defined in the ITU recommendations. [91] Deal with the design of AAL protocols to be used for data communications. Protocols to be used on top of the AAL are dealt with in [26].
Finally, performance studies related to the design of these protocols have been presented in [10] and [30].
An environment where the use of a wireless service on ATM can be expected to become prevailing is the Internet. The use of ATM for the transfer of packets for the Internet Protocol (IP) is being studied ([17], [62], [99]).
Many papers have been published on architectural aspects of providing a wireless service using an ATM network ([5], [6], [26], [37], [66]). Besides describing entities and protocols, most of these papers also identify different options for providing a wireless service, such as the direct method vs. the indirect method, and message mode vs. streaming mode (see Chapter 2).
A significant contribution to the definition of protocols and interfaces between entities has been made by research projects in the RACE (Research and development in Advanced Communications technologies for Europe) program.

Relatively few papers report on the implementation of CLSs, i.e., the nodes installed in the network to route packets. [25] Describe the design and implementation of a large scale wireless server, connecting multiple ATM links, at the Alcatel Bell Telephone Research Centre. Design of small wireless servers, attached to a single ATM link is described in [6], [63], and [91].
The problem of the use of ATM connections and the reservation of bandwidth on these connections has received considerable interest. This has resulted in a large number of publications on the subject. A large number of these publications are authored by Gerla et

al., and deal with a so-called bandwidth advertising algorithm ([25], [27], [40], [41], [42]). Others deal with mechanisms to modify bandwidth reservations during the lifetime of a connection on an on-off basis ([37]). Proposals to do this on a multilevel basis have also been done ([4], [29], [30], [97], [106]). Most of these papers contain also some performance analyses of proposals.

1.4 Summary and Concluding Remarks

In this chapter, we have defined the notion of wireless communications, as opposed to connection-oriented communications. Further, the reader has been introduced to the B-ISDN, a public telecommunication network aimed at providing a large base of applications to it users via an integrated access. Eventually, the communication needs of all these applications must be served by a single underlying infrastructure based on ATM. The basics of ATM are asynchronous time division multiplexing and fast packet switching.

ATM will also be used to provide a wireless service to the users. Applications using this service have been identified, and basic requirements have been outlined. Most end-systems with these applications will access the B-ISDN via LANs or MANs. Throughput requirements of individual applications will range from less than a kilobit per second to values of the order of a gigabit per second.

A survey of related work that can be found in the literature has been given.

We have seen that ATM is a connection-oriented technique, where a connection is established prior to transferring fixed-size cells. Connection establishment requires knowledge about the traffic that is going to be transferred, in order to determine the bandwidth to be reserved. There is clearly a mismatch between ATM and the wireless service some applications require. The wireless service should transfer data units of any size. Moreover, it is essential to wireless communications that no advance knowledge about the traffic to be transferred is exchanged with the system that provides the service. To bridge this gap between ATM and the wireless service to be provided is the aim of this Project.

CHAPTER TWO

ARCHITECTURAL FRAMWORK

The design of a complex communication system, such as the one this Project is concerned with, is a very complicated task. In order to be able to cope with this complexity, design methodologies have been developed, which allow for recursive decomposition of a system into subsystems that exhibit a lower complexity. This chapter will use modeling concepts, in order to come up with an architectural framework for the provision of wireless communications over an ATM network. The architectural framework allows us to put the protocols that have (partially) been designed by different organizations in perspective. The various entities that are needed to provide the service are identified, their functionality is defined, and the way they interact with each other is specified.

We base our work on the above mentioned design methodology, because it uses a number of clear, well-defined concepts. Furthermore, the equivalence of the system in different stages of the design is more easily seen than with other methodologies,
e.g., the one described in Recommendation I.130 of the ITU ([12]). It also provides a consistent and rigorous basis for the framework. Let us introduce the concepts used in the architectural framework..It should be noted that these concepts may be defined differently from what the user is familiar with, and that terms that are used to denote the concepts are often also used in a different context, with a slightly different meaning.

The first concept to be introduced is that of a service. A service specification is a description of the (required) external behaviour of a system. The (communication) system, or service provider, is regarded as a whole, no component parts nor are their mutual interactions visible. Users interact with the service provider at
Service Access Points (SAPs). A service can be described by specifying these interactions, or service primitives, their parameters, and the possible sequences in which the primitives can be issued at the different SAPs. The service description is at the highest level of abstraction. It does not prescribe how the interactions should take place.

A protocol specification is a description of a system as a collection of system parts.
A protocol is a set of rules according to which subsystems (or protocol entities) cooperate to collectively provide a service. For this purpose, the protocol entities use the service provided by an underlying service provider.
By recursively decomposing a service provider in a protocol and an underlying service provider, a complex service can be decomposed into a number of layered protocols, until an underlying service is encountered that is already available.

A result of this recursive decomposition is a network architecture. It describes the protocol entities that can be identified in the system, the way they interact, and how they cooperate to provide certain services.
Let us finally introduce the notion of a real system. A real system is the implementation of a service provider or (a set of) protocol entities, i.e., compositions of software and hardware.
The architectural framework that is presented in this chapter defines the service that is required from the system under design, i.e., the system providing a broadband wireless service to its users. It also describes a service that is already available for use in the design of this system, i.e., the service provided by an ATM network. Furthermore, the network architecture identifies intermediate services, the protocol entities that provide either the required or an intermediate service, and the protocols.
It should be noted that the protocols described in this chapter are not novel. They have been standardized, or proposed for standardization. However, the overall architectural framework in which the protocols are positioned is new.
The descriptions of the services and protocols are given in prose. It is beyond the scope of this Project to give formal specifications.

In Section 2.1; the service description of the system to be designed will be given.

In Section 2.2, the service provided by an ATM network will be described. The actual decomposition is performed in Section 2.3, which is the main body of this chapter. In this section, network architecture for providing a broadband wireless service using ATM is presented. Section 2.4 and Section 2.5 come up with candidate protocols for the protocol layers identified in the network architecture, the ATM Adaptation Layer (AAL), and the Broadband Wireless Network Layer. This chapter ends with summary and concluding remarks, in Section 2.6.

2.1 The Broadband Connectionless Service (BCLS)

This section specifies the service that should be provided by a system for wireless data communications over ATM. The service will be referred to as Broadband Connectionless Service (BCLS). The definition of the BCLS in the following subsections covers those aspects of the service that are relevant for this Project, given the level of detail of the design we are aiming at. It captures requirements that are expected to be posed on systems for broadband wireless data communications. The service description functions as a starting point for the design in this Project.

Specifications of systems for broadband wireless data communications have been given elsewhere. Bellcore specifies the Switched Multi-megabit Data Service (SMDS) in the Introduction. General features of SMDS and protocols for the interface to an SMDS system are defined. However, a service description as given here is not available for SMDS. The ETSI specifies the Wireless Broadband Data

Service (CBDS) in [32], [33], [34], and [35]. The CBDS is very similar to SMDS, and so is the specification. The ITU specifies a Broadband Wireless Data Service (BCDS) in Recommendation F.812 ([15]), and some guidelines for the support of this service by B-ISDN in Recommendation I.364 ([80]). The service specification of Recommendation F.812 specifies the interface to a system providing a BCDS over B-ISDN in general terms. Recommendation I.364 specifies a protocol for this interface (the UNI). Both recommendations are not yet complete.

The service description presented here is based on [50]. Most of the information has been taken from the SMDS specification, since the ITU recommendation on BCDS is far from complete. Conflicts of this service specification (and the SMDS specification) with the ITU recommendations will be explicitly mentioned.

We only describe the observable behavior of the system, without describing how a UNI should be implemented. This is discussed later in this chapter, since we believe that it should be considered in a later design stage. Where necessary, the observable behaviour has been derived from the interface specification in introduction and [80].

In Section 2.1.1 we describe general aspects of the service. The service primitives that have been defined are presented in Section 2.1.2. The issue of QoS is discussed in Section 2.1.3. Section 2.1.4 gives some extra features of the service, called supplementary services, and Section 2.1.5 gives values for the QoS parameters.

2.1.1 General

The BCLS provides a means by which Service Data Units (BCL-SDUs) of variable but limited length are delimited and transparently transferred from one source Service Access Point (BCL-SAP) to one or more destination BCL-SAPs in a single service access, without establishing or later releasing a connection between source and destination BCL-SAPs.

Associated with the service are certain Quality of Service (QoS) parameters. It is assumed that the values of these parameters are fixed at the time of subscription to the service, or negotiated by management procedures. Alternatively, values for some of the parameters could be determined per transferred BCL-SDU. This option has been defined in the ITU BCDS. SMDS on the other hand does not allow this. We assume in this Project that QoS cannot be selected on a per- SDU basis, but only per service subscription.

The BCLS can also provide some extra features; called supplementary services (see Section 2.1.4). The provision of these features is negotiated at the time of subscription to the service.

2.1.2 Service Primitives

Two service primitives are needed to provide the BCLS:

• BCL-DataRequest, with parameters

Source Address, Destination Address, and BCL-SDU; and

• BCL-DataIndication, with parameters.

Source Address, Destination Address, and BCL-SDU.

The BCL-SDU parameter, which contains the user data, may have any length up to a certain maximum. This maximum length has been defined to be 9188 octets.

The Source Address parameter refers to the BCL-SAP at which the BCL-DataRequest primitive is issued. The Destination Address parameter refers to either an individual BCL-SAP or a number of BCL-SAPs in case of multicast.

In general, a BCL-DataRequest primitive at a certain BCL-SAP, corresponding to the Source Address parameter, will result in a BCL-DataIndication primitive at the BCL-SAP corresponding to the Destination Address parameter. If the Destination Address parameter refers to a number of BCL-SAPs, the BCL-DataRequest primitive will result in BCL-DataIndication primitives at all BCL-SAPs (multicast).

The value of all three parameters will be the same for a BCL-DataRequest primitive and the corresponding BCL-DataIndication primitive(s). For a given source-destination pair of BCL-SAPs, a BCL-DataRequest primitive will result in a BCL-DataIndication primitive before any subsequent BCL-DataRequest primitive results in a BCL-DataIndication primitive, i.e., there is in-sequence delivery.

With a certain (low) probability, the BCL service provider may exhibit a slightly different behaviour. The value of this probability depends on the agreed QoS. The following cases of exceptional behaviour are identified:

• Loss

The BCL-DataRequest primitive does not result in a BCL-DataIndication primitive.

• Duplication

The BCL-DataRequest primitive results in two BCL-DataIndication primitives at the same BCL-SAP.

• Misdelivery

The BCL-DataRequest primitive results in a BCL-DataIndication primitive at a

BCL-SAP not specified in the destination address parameter.

• Corruption

The BCL-SDU parameter of a BCL-DataIndication primitive differs from the BCL-SDU parameter of the corresponding BCL-DataRequest.

• missequencing

For a given source-destination pair of BCL-SAPs, a BCL-DataRequest primitive results in a BCL-DataIndication primitive which happens later than the BCLDataIndication primitives corresponding to one or more subsequent BCL-DataRequest primitives.

2.1.3 Quality of Service

The term Quality of Service (QoS) refers to certain characteristics of a wireless transfer as observed between the BCL-SAPs. QoS parameters will be fixed at the time of subscription or negotiated by management procedures during service provision.

The following QoS parameters can be identified:

• Maximum throughput

The maximum number of bits (contained in BCL-SDUs) that may be offered by a service user to the service provider per time unit.

• Transit delay

The elapsed time between BCL-DataRequest primitives and the corresponding BCL-DataIndication primitives.

• Loss probability

The probability that a loss occurs, estimated by the ratio of lost BCL-SDUs to the total number of transferred BCL-SDUs.

• Duplication probability

The probability that a duplication occurs, estimated by the ratio of duplicated BCL-SDUs to the total number of transferred BCL-SDUs.

• Misdelivery probability

The probability of misdelivery, estimated by the ratio of misdelivered BCL-SDUs to the total number of transferred BCL-SDUs.

• Corruption probability

The probability that a corruption occurs, estimated by the ratio of corrupted

BCL-SDUs to the total number of transferred BCL-SDUs.

• missequencing probability

The probability of missequencing, estimated by the ratio of missequenced BCL-SDUs to the total number of transferred BCL-SDUs.

2.1.4 Supplementary Services

The term Supplementary Services refers to some extra features that can be provided by the BCLS to extend its functionality. The provision of these supplementary services will either be fixed at the time of subscription, or negotiated by management procedures during service provision.

The following supplementary services have been identified:

• Closed User Group (CUG)

This supplementary service enables users to form groups, to and from which access is restricted. A specific user may be a member of one or more closed user group. Closed user groups may have additional capabilities whereby individual members may send data to users outside the group, and/or receive data from users outside the group, e.g., in Virtual Private Networks (VPNs).

• Address Screening

This supplementary service ensures that a user cannot or can only receive data from certain users and that a user cannot or can only send data to certain users.

Address screening may be applied to individual addresses as well as to group addresses. Source address screening imposes restrictions on the set of users that are allowed to send to a specific destination. Destination address screening restricts the set of users to which a certain source may send. Address screening may be used for security reasons. It can also be used to create closed user groups.

2.1.5 QoS Values

As stated in Section 2.1.3, values of QoS parameters are not a part of this service specification. However, in order to have target values that can be used in performance studies, we list some values as they are specified for SMDS in the introduction in Table 2-1

Table 2-1: Values for QoS Parameters in SMDS

QoS parameter	value
maximum throughput	4 - 34 Mbits/s
transit delay (single destination)	95% within 20 - 140 ms (depending on the type of UNI)
transit delay (multiple destinations)	95% within 100- 220 ms (depending on the type of UNI)
loss probability	$1x10^{-4}$
duplication probability	$5x10^{-8}$
misdelivery probability	$5x10^{-8}$
corruption probability	$5x10^{-13}$
missequencing probability	$1x10^{-9}$

2.2 The ATM Service

Let us now describe the service provided by an ATM network. In the B-ISDN Protocol Reference Model (see Section 2.1.2) and in the protocols defined for B-ISDN, a distinction is made between a user plane and a control plane1. The user plane contains the functionality for transferring information on behalf of the user, given the presence of appropriate connections. The control plane contains functionality for the establishment, modification, and release of these connections.

The functionality of the control plane is comprised in the signaling system.

For the transfer of BCL-SDUs, we are interested in the composite behavior of the ATM layer in the user plane and the part of the signaling system controlling this ATM layer ([76]). A user of an ATM network willing to transfer data, should first set up an ATM connection via the control plane, transfer the data via the user plane, and finally release the connection again via the control plane. Within the ITU recommendations, no description of the composite behavior of user plane and control plane is provided. A first proposal for such a description has been given in [56]. However, this proposal describes the behavior of a B-ISDN system at a much higher layer, i.e., at the layer where multi-media services are supported.

Two major reasons prevent us from coming up with an ATM service description specifying the composite behavior of user and control plane. The first one is that in B-ISDN no distinction is made between control of the ATM layer and control of higher layers, so that it is somewhat artificial to attribute part of the control plane to the ATM service. The second one is that the specification of control protocols for B-ISDN is still an ongoing process, where the functionality of the protocols is gradually extended by defining protocols for different capability sets ([84], [85]).

Our objective with respect to ATM is to describe the external behavior of a given system, not to specify the requirements on a system under design. Therefore, the description provided here gives general characteristics of the ATM service as far as control aspects are concerned, and a more detailed specification for the transfer of user information over ATM connections.

In Section 2.2.1 we describe some general aspects of the ATM service. Characteristics of the service, related to the control of connection, are given in Section 2.2.2.

The service primitives that have been defined for the transfer of user information are given in Section 2.2.3. Related to a certain ATM connection are traffic parameters and QoS parameters, describing characteristics of the traffic generated by the user, and the way the ATM service transfers the traffic. These are discussed in Section 2.2.4 and Section 2.2.5. Finally, in Section 2.2.6, QoS parameters related to the control of connections are discussed.

2.2.1 General

The ATM service provides a means by which ATM Service Data Units (ATM-SDUs) of a fixed, standardized length are delimited and transparently transferred from a single source ATM-SAP to one or more destination ATM-SAPs along a previously established connection. Establishment and release of a connection can be done either on demand or (semi-)permanently by management procedures.

Associated with the service are certain QoS parameters. The values of these parameters are negotiated during the connection establishment. Furthermore, traffic parameters are associated with an ATM connection, specifying characteristics of the traffic a user offers to the network for transfer over the connection.

These parameters are negotiated during connection establishment and can be renegotiated during the lifetime of the connection.

2.2.2 Control of ATM Connections

ATM-SDUs can only be transferred from a source ATM-SAP to one or more destination

ATM-SAPs if an ATM connection from the source SAP to the destination SAP(s) is available. A service user has access to the connection at an ATM Connection Endpoint (ATM-CEP). The presence of the connection corresponds to the presence of a source CEP and one or more associated destination CEPs. An ATM connection has a source CEP in the source SAP, and a destination CEP in each destination SAP.

Two types of connections have been identified, Virtual Channel Connections (VCCs) and Virtual Path Connections (VPCs). An ATM-SDU transferred over a VPC has associated with it an extra identifier (the VCI), which can be used by the user of the ATM service to identify the SDU for its own purposes. The VCI is transferred transparently over a VPC.

Associated with connections of both types are QoS parameters (Section 2.2.5), and a set of traffic parameters (Section 2.2.4). Both give characteristics of the communication over the connection, and are the result of negotiation between the service users and the ATM network.

ATM connections can be available on demand, or on a (semi-)permanent basis. The installation of (semi-)permanent connections (either VCC or VPC) is taken care of by management. The traffic parameters of a (semi-)permanent connection can be negotiated and changed during the lifetime of the connection.

The signaling system of the ATM network is responsible for establishing on demand connections. During the establishment, which is initiated by one of the service users, QoS parameters and traffic parameters are negotiated. The traffic parameters can be renegotiated during the lifetime of a connection, the QoS can not.

QoS parameters are also associated with the signalling itself. They give characteristics of the control of connections. These are elaborated on in Section 2.2.6.

2.2.3 Service Primitives

Service primitives are always associated with a single connection. They are issued at a certain CEP in a SAP. The following two service primitives have been defined for the transfer of user information over the ATM service:

• ATM-DataRequest, with parameters
ATM-SDU, Submitted Loss Priority, Congestion Indication, ATM-user-to-ATM-user Indication, and VCI (only for a VPC); and

• ATM-DataIndication, with parameters ATM-SDU, Congestion Indication, ATM-user-to-ATM-user Indication, and VCI (only for a VPC).

The ATM-SDU parameter, which contains the user data, has a length of exactly 48 octets. It is passed transparently by the ATM network from source to destination(s) along the ATM connection. The Submitted Loss Priority parameter indicates the importance of the ATM-SDU relative to other SDUs transferred over the same connection. It is either high or low. The Congestion Indication parameter is used to indicate that the ATM-SDU has passed through a congested network node. Finally, the ATM-user-to-ATM-user Indication (AUU) parameter is transparently passed by the ATM network, just like the ATM-SDU. Its length is a single bit. The VCI is only a parameter if the CEP where the primitive is issued is the endpoint of a VPC. It is passed transparently by the ATM network, and can be used by the service user as an extra identification for the ATM-SDU. Given the availability of a connection, the general operation of the ATM service is as follows. An ATM-DataRequest primitive at a certain source CEP in a SAP will result in an ATM-DataIndication primitive at the destination CEP associated with the source CEP. If several CEPs are associated with the source CEP, an ATMDataIndication primitive will result at all destination CEPs. The ATM-SDU and the AUU parameters are identical for the ATM-DataRequest and the corresponding ATM-DataIndication(s). If the Congestion Indication Parameter was not set in the request primitive, it may be either set or not set in the indication primitive. If it was set in the request primitive, it will be set in the indication primitive as well. For a given pair of CEPs, the sequence of ATM-DataIndication primitives is identical to the sequence of the corresponding ATM-DataRequest primitives.

With a certain (low) probability, the ATM service provider may exhibit slightly different behaviors. The value of this probability depends on the QoS agreed for the connection. The following cases of exceptional behaviors are identified:

• Loss

The ATM-DataRequest primitive does not result in an ATM-DataIndication primitive.

• misinsertion

An ATM-DataIndication primitive occurs at a CEP, while no ATM-DataRequest has been issued at an associated source CEP.

• Corruption

The ATM-SDU parameter of an ATM-DataIndication primitive differs from the ATM-SDU parameter of the corresponding ATM-DataRequest.

2.2.4 Traffic Parameters

Traffic parameters are used to describe traffic characteristics of an ATM connection.

They characterize the traffic that is offered by the service user to the network. These parameters are negotiated during the establishment of a connection, and may be renegotiated during the lifetime of a connection.

Currently the ITU specifies only a single traffic parameter:

• Peak cell rate

The peak cell rate is defined as the inverse of the minimum time elapsing between two consecutive ATM-DataRequest primitives.

Other parameters, such as the mean cell rate, are being studied.

2.2.5 Quality of Service of an ATM Connection

A number of QoS parameters are associated with an ATM connection. They can be negotiated during the establishment of the connection. The agreed QoS parameters will only be met by the network if the user confines to the agreed traffic parameters. Up to now only two QoS parameters have been identified within the ITU. They, however, have not yet been defined exactly:

• Cell delay variation

The cell delay variation is the deviation of the actual cell delay from the mean cell delay experienced for the connection.

• Cell loss ratio

The cell loss ratio is the ratio of lost ATM-SDUs to the total number of transferred ATM-SDUs on a connection. For a single ATM connection, two cell loss ratio objectives can be defined, one for ATM-SDUs with a high submitted loss priority parameter and one for those with a low submitted loss priority parameter.

2.2.6 Quality of Service of the Signaling System

Other relevant QoS parameters for the ATM service are those characterizing aspects of the control of connections. These have not been defined by the ITU yet.

In [64], a method for developing performance measures related to these parameters has been proposed.

We are only interested in QoS parameters quantifying behavioral aspects of the control plane as far as the control of ATM is concerned. For the performance studies in this Project, we need to make assumptions about the following parameters:

• Connection establishment delay

The connection establishment delay is the time elapsing between the requests of a user to the signaling system to establish a connection, and the confirmation from the signaling system that the connection is available.

• Traffic contract renegotiation delay

The traffic contract renegotiation delay is the time elapsing between a request from a user of a connection to modify the traffic contract, and the confirmation from the network that the new contract is in force.

2.3 Network Architecture

Now that we have defined the Broadband Wireless Service in Section 2.1, we have the starting point for the decomposition of the system under design in subsystems. The network architecture that results from this decomposition specifies the types of protocols that are needed to provide the BCLS, the protocol entities that implement the protocols, and the underlying services that are used by these entities. We present the network architecture in a stepwise approach, where the steps correspond to subsequent decompositions of the service. In the past, similar decomposition techniques have been successfully applied to other problems.

In this section, we will first introduce the notation that is used to present the network architecture, in Section 2.3.1. Basically, there are two methods of providing a wireless service over ATM. We introduce them in Section 2.3.2.

The subsequent decompositions of the BCLS are presented in Section 2.3.3 for the first method, and in Section 2.3.4 for the second method. An overall view of the network architecture for both methods will be given in Section 2.3.5. In Section

2.3.6, we will derive a protocol reference model from these network architectures.

Finally, in Section 2.3.7, we relate the architectural framework given in this chapter to architectures and models defined in standards.

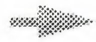
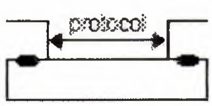
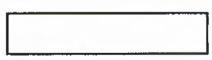
2.3.1 Notation

Let us first explain the notation that is used to describe the network architecture in a symbolic way (see Table 2-2). We present the network architecture by showing the decomposition steps that are taken. In each figure (Figure 2-1 through 2-12) both the unpartitioned and the partitioned (sub) system are shown.

The unpartitioned (sub) system is shown on the left, the (sub) system after the decomposition step is shown on the right. The gray arrow (a) denotes this decomposition step. A system is shown in two different representations, one in the upper part of the figure, and one in the bottom part. The bottom part represents the actual network architecture by displaying protocols, subsystems, protocol entities, and underlying services. The upper part is displayed for illustration. It gives a representation of a real system that can be associated with the network architecture. Dashed lines (b) show the correspondence between the two representations.

For the description of the network architecture we use the following notation. A (sub) system of which only the external behaviour is considered (a service provider), is

Table 2-2: Notation for the Symbolic Description of Network Architecture

	(a): decomposition step
	(b): corresponding representations
	(c): service provider
	(d): cooperating protocol entities
	(e): protocol conversion
	(f): end-system
	(g): interconnection
	(h): intermediate system
	(i): network

represented by a rectangle (c). The black ovals denote the SAPs at which service users interact with the service provider. The cooperation between a numbers of subsystems (e.g., protocol entities) according to the rules of certain protocol, using an underlying service provider, is represented by a double arrow (d). Adjacent rectangles are consider as subsystems that can directly interact.

Subsystems are assumed to cooperate with other subsystems using a single protocol. If they are not, i.e., if some protocol conversion has to be performed, the subsystem is drawn with diagonal lines in it (e).

For the representation of a real system, four symbols will be used. They represent real systems that are end-systems from the point of view of the BCLS (f), interconnections at a lower layer (g), intermediate systems (h), and networks (i).

2.3.2 Methods of Providing a Wireless Service

Within the ITU, two methods have been identified to provide a wireless service using an ATM-based B-ISDN ([72]). These have been characterized as follows:

• "Indirectly via a B-ISDN connection-oriented service:

In this case a transparent connection of the ATM layer, either permanent, reserved or on demand is used between B-ISDN interfaces."

• "Directly via a B-ISDN wireless service:

In this case the wireless service function would be provided within the B-ISDN."

The indirect method uses end-to-end ATM connections to connect a pair of users that want to communicate wireless. The direct method use special nodes in the B-ISDN, called Wireless Servers (CLSs), to which users that want to communicate wireless are connected by means of ATM connections. These two methods of providing a wireless service over ATM are not mutually exclusive. Both methods can be used simultaneously over the same ATM network. Furthermore, a single user can use both methods, e.g., depending on the destination.

The two ways of providing a wireless service are reflected in the network architecture. Let us first present the decomposition of the BCLS that corresponds to the indirect method.

2.3.3 Indirect Method

The BCLS can be provided on top of an end-to-end connection-oriented (CO) network (Figure 2-1). A protocol is needed that relates the required destination address for a BCL-SDU to a SAP of the underlying network, and routes a PDU containing the

SDU on a connection to that SAP. A connection management function is needed in the protocol to request the underlying network to establish, maintain and release end-to-end connections between the protocol entities in the end-systems when needed. This protocol will be referred to as Broadband Wireless Network Protocol (BCL NP).

Note that the term end-system refers to a real system that is an end-system with respect to the B-ISDN, i.e., to a node that is connected to the B-ISDN via its User Network Interface (UNI). According to the B-ISDN Reference Configuration, described in [82], this is a Terminal Equipment (TE) or Network Terminator type 2 (NT2). However, it can be an intermediate system in an internet, e.g., an interworking unit to a LAN or MAN.

Figure 2-1: Decomposition of the BCLS using the Indirect Method

The end-to-end CO network should provide a means by which SDUs of a variable size can be transferred between end-systems over a previously established connection. In an ATM environment, this service can be provided by an ATM Adaptation Layer (AAL) protocol, using the ATM service as the underlying service (Figure 2-2). Such an AAL protocol is also an end-to-end protocol. It segments SDUs of a variable length into fixed size ATM-SDUs, and reassembles the variable size SDUs again at the destination.

Figure 2-2: Decomposition of the End-to-end CO Network

The ATM service has been described in Section 2.2. Although the ATM service is taken as a starting point for the design, we include its decomposition in the network architecture. This is to obtain an overall framework of protocols and entities needed for the provision of the BCLS and to be able to identify which protocols need to be implemented in which real system.

The ATM service is provided by an ATM network to which end-systems have access over a UNI (Figure 2-3). This UNI is indeed the one that has been specified for the B-ISDN. The ATM protocol to be used over this UNI has been specified in [76]. the underlying Physical Layer (UNI PL) service has been defined in [13].

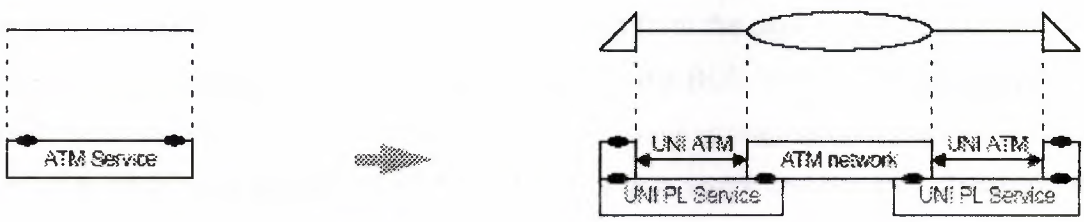


Figure 2-3: Decomposition of the ATM Service

The ATM network itself can be seen as composed of ATM nodes, which are in fact the exchanges of the B-ISDN. Within the ITU, the interface between these exchanges is called the Network Node Interface (NNI). The ATM nodes cooperate according to the ATM protocol that has been defined for the NNI ([76]). For their communication, they use the service provided by the physical layer, as it has been defined for the NNI (NNI PL) (Figure 2-4). Note that only in some of the ATM nodes (local exchanges) interworking with UNI protocols has to take place. The other nodes (transit exchanges) just have to implement NNI ATM protocols.

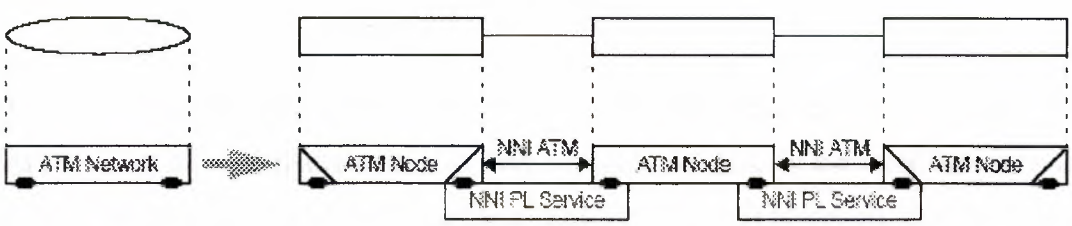


Figure 2-4: Decomposition of the ATM Network

2.3.4 Direct Method

In the network architecture presented in the previous section, the BCL network protocol was only implemented in the end-systems. In the network architecture that will be presented now, the network will also implement this protocol. This network architecture corresponds to the so-called direct method of providing a wireless service.

If the direct method is used, the BCLS will be provided to the user by some functionality in an end-system, which is cooperating with the BCL Network (Figure 2-5). The end-system and the BCL Network cooperate according to a BCL Network Access Protocol (BCL-NAP) using a BCL Access Network as an underlying service. The protocol entity in the end-system will pack a BCL-SDU, received from the user, in a PDU together with the source and destination address, and pass it to the BCL Network. Subsequently, the BCL Network will transport the PDU, and pass it to the appropriate remote user entity. In order to enable the passing of PDUs between a user entity and the BCL Network, the BCL Access Network provides for the communication between the cooperating entities.

This access network can be connection-oriented or wireless. If it is connection-oriented, a connection between the user entity and the BCL Network needs to be established before communication is possible.

Not all users will access the BCL Network using the same BCL-NAP, and using the same protocols for the BCL Access Network. The access network can for instance be based on

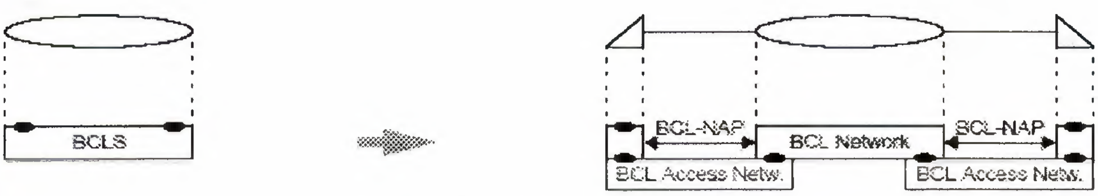


Figure 2-5: Decomposition of the BCLS using the Direct Method

ATM, or on the DQDB protocols ([65]). In this heterogeneous environment, the BCL-NAPs will not be exactly the same, but they will provide roughly the same functionality, e.g., the same addressing conventions have to be used by both protocols. For that reason, PDUs from one protocol can be mapped on PDUs from the other, quite easily. Figure 2-6 represents the decomposition of the BCLS in case of heterogeneous access networks.

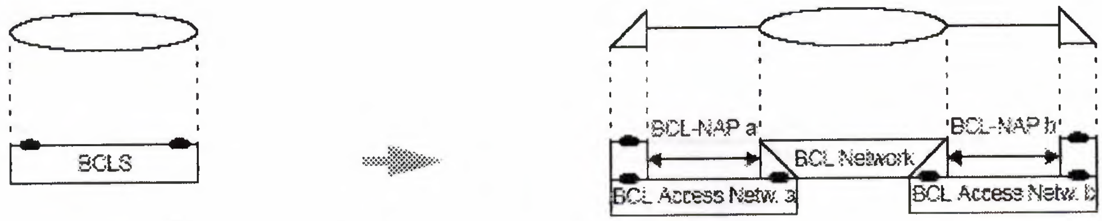


Figure 2-6: Decomposition of the BCLS with Heterogeneous Access Networks

Let us first consider the access network to the BCL network. Two candidates for the access protocol are ATM and DQDB. The choice for a particular underlying protocol has an impact on the structuring of the access network. An underlying protocol system can be thought of as a predefined building block, with a prede- fined functionality. The other building blocks have to match this functionality.

Since this Project is concerned with the transfer of wireless data, using ATM, we will focus on BCL Access Networks that have ATM as an underlying protocol. Such an access network has a connection-oriented nature.

Figure 2-7: Decomposition of the BCL Access Network

The BCL Access Network should be able to transfer BCL-NAP PDUs of a variable size. In order to adapt to the fixed size of ATM-SDUs, the BCL Access Network is decomposed into cooperating protocol entities, taking care of segmentation and reassembly of BCL-NAP PDUs, and an underlying ATM service (Figure 2-7). The protocol used for segmentation and reassembly is an AAL protocol.

The ATM service provider is very similar to the ATM service defined for the indirect method of providing wireless services. The difference here is that the access of the BCL Network to the ATM network is through an NNI instead of through a UNI. This yields the following decomposition (Figure 2-8).

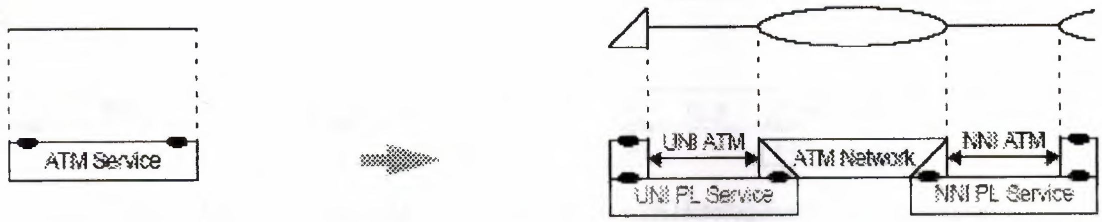


Figure 2-8: Decomposition of the ATM Service for the BCL Access Network

Now that we have presented the decomposition of the access network to the BCL Network, let us present the decomposition of the BCL Network itself. A BCL Network consists of a number of protocol entities, which cooperate according to a Broadband Wireless Network Protocol (BCL NP), and which communicate with each other using the service of an Interconnection Network (Figure 2-9).

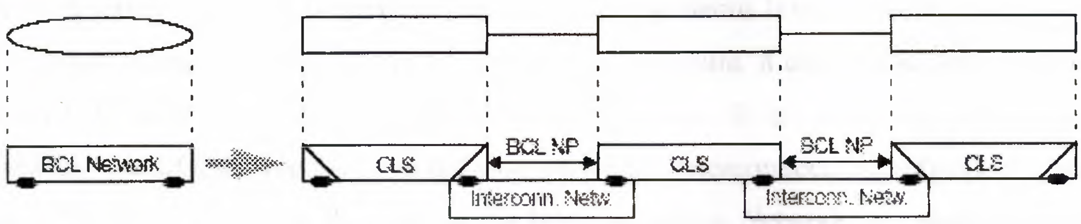


Figure 2-9: Decomposition of a BCL Network

These protocol entities route PDUs containing a BCL-SDU including source and destination address from source to destination. The PDUs are transferred between two entities along a connection established between SAPs in the Interconnection Network. The establishment and maintenance of connections, in order to connect protocol entities, is also a function of the BCL NP. A real system that implements the protocol entity of the BCL NAP and/or BCL NP together with the protocol entities of underlying protocols, is called a Wireless Server (CLS). Two types of CLSs can be identified depending on whether or not it directly communicates with an end-system. An Access CLS does communicate directly with end-systems, and should thus implement both the BCL NP and the BCL NAP and perform conversion between the two. A Transit CLS does not, and

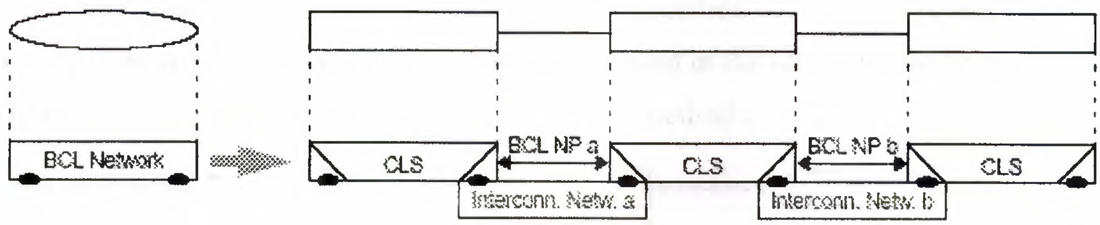


Figure 2-10: Decomposition of a BCL Network with Heterogeneous Interconnection Networks

should only implement the BCL NP. Transit CLSs will normally only be present in very large networks. The Interconnection Networks that are identified in the last decomposition step are not necessarily of the same type. One could for instance be based on ATM while another is based on DQDB. This would imply that the service provided by the Interconnection Network is not the same, and thus that the BCL NPs used over this service are slightly different to adapt to those changes. Figure 2-10 shows the decomposition of the BCL Network in case of heterogeneous Interconnection Networks. The Interconnection Network can be very diverse in nature. It can be a simple point-to-point link, with a datalink protocol running on top of it. It can also be a multi-access network like DQDB. We focus on the case where the Interconnection Network is based on ATM. We can further decompose the Interconnection Network into entities taking care of segmentation and reassembly, using the ATM Adaptation Layer protocol, and an underlying ATM service for the transfer of cells (Figure 2-11).

Figure 2-11: Decomposition of an ATM-based Interconnection Network

The ATM service is very similar to the ATM service defined for the BCL Access Network. Here all users of the ATM service access the ATM Network via an NNI.

This yields the following decomposition (Figure 2-12).

The ATM network that is obtained from the decomposition of the BCL Access Network and the Interconnection Network, can be decomposed in the same way as the one that is obtained during the decomposition for the indirect method of wireless service

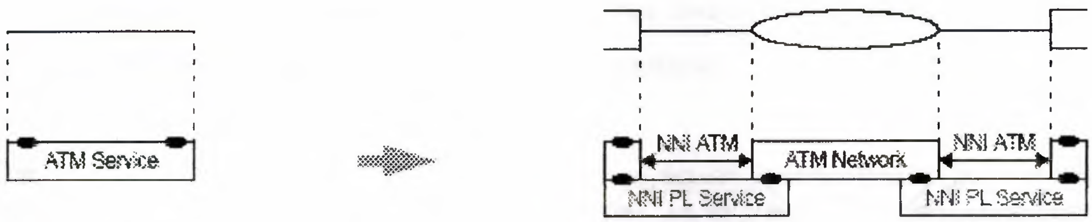


Figure 2-12: Decomposition of the ATM Service for the Interconnection Network

provision. These ATM networks can in fact be realized by the same real network. Thus, ATM connections can be used by end-systems to access the BCL Network and by CLSs to connect to each other. In fact, the same real network can also realize the ATM network identified in Section 2.3.3, thus allowing for the coexistence of the direct and indirect method of providing wireless services over the same ATM network.

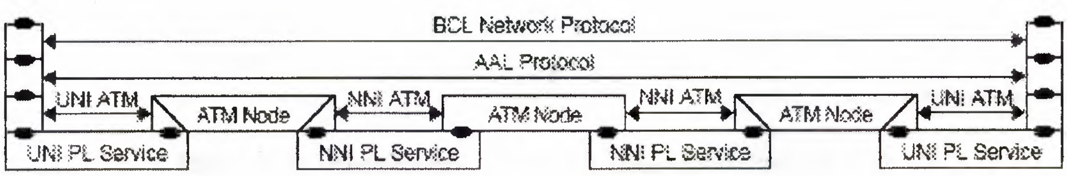


Figure 3-13: Network Architecture for the Indirect Method

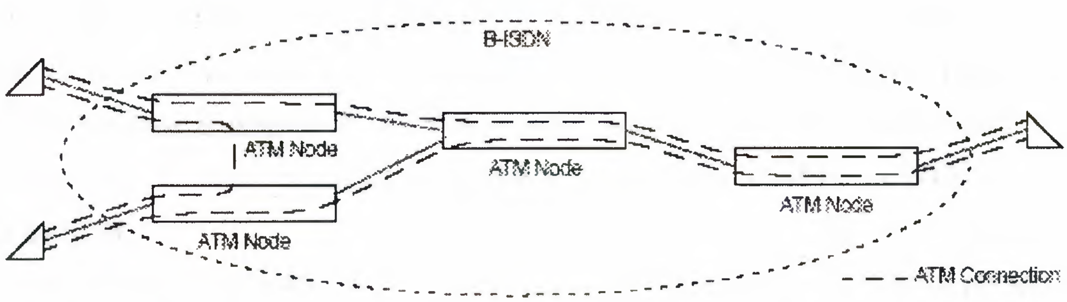


Figure 2-14: Example Network Configuration for the Indirect Method

2.3.5 Overall View on the Network Architecture

Let us try to summarize the results from the previous sections, in order to obtain an overall view of the network architecture. We present two figures that put all the mentioned protocols in perspective, one for the indirect method, and one for the direct method. Furthermore, we illustrate the network architectures by giving possible configurations of real systems that provide the wireless service.

In case of the indirect method, the functionality of the BCL NP and the AAL protocol is only present in the end-systems (Figure 2-13). The intermediate systems contain only ATM and Physical Layer functionality. As a consequence,

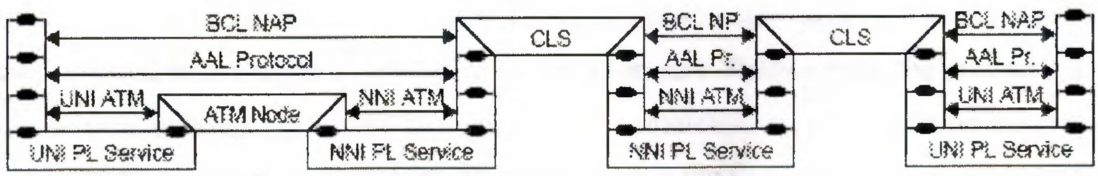


Figure 2-15: Network Architecture for the Direct Method

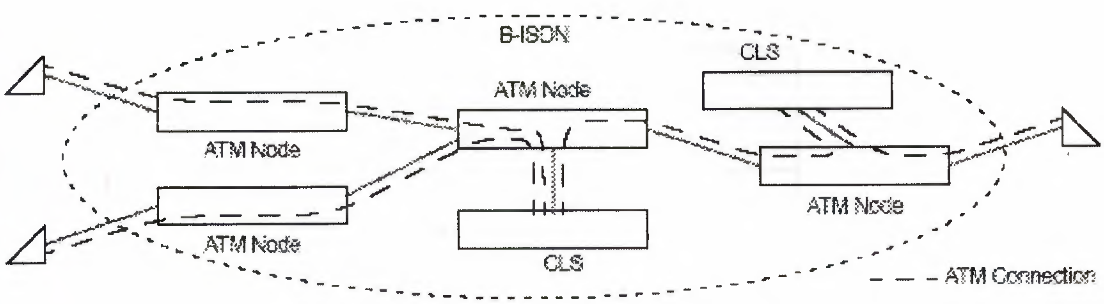


Figure 2-16: Example Network Configuration for the Direct Method

only ATM nodes are present in the network. Figure 2-14 shows a possible network configuration of an ATM-based B-ISDN, where the indirect method of providing a wireless service is employed. All end-systems that wish to communicate need to be connected by ATM connections. These connections are switched by the ATM nodes in the B-ISDN.

In case of the direct method, BCL NP and AAL protocol functionality is present in intermediate systems within the B-ISDN, i.e., in CLSs (Figure 2-15). The CLSs communicate via ATM nodes, according to the BCL NP. Entities in the endsystems cooperate with a CLS according to a BCL NAP, probably also via ATM nodes. Figure 2-16 shows a possible configuration for providing a wireless service using the B-ISDN according to the direct method. The two CLSs in this figure are connected by means of an ATM connection. End-systems that wish to communicate use an ATM connection to a CLS to transfer their PDUs.

2.3.6 Protocol Reference Model

In the network architectures, presented in the previous sections, a number of protocols have been identified for different parts of a system providing a BCLS.
Some of these protocols have much in common, their major difference being whether they are used for access to a network or within a network. Furthermore, the protocols use the service provided by other protocols to communicate. Here, we define a layered model, a protocol reference model. Protocols with a similar functionality reside in the same layer. Protocols that are used by protocols from a specific layer for their communications reside in the next lower layer.

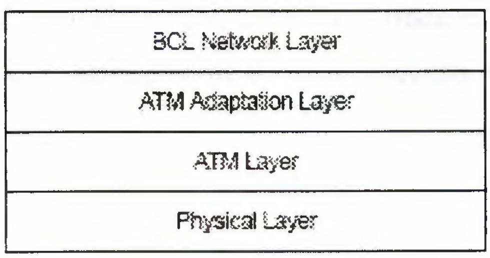


Figure 2-17: BCLS Protocol Reference Model

Figure 2-17 shows the BCLS Protocol Reference Model. The following four layers can be identified: BCL Network Layer, ATM Adaptation Layer, ATM Layer, and Physical Layer. Below we will discuss these layers in detail.

BCL Network Layer

The BCL Network Layer contains those protocols that are concerned with the end-to-end transfer of individual BCL-SDUs between source and destination.

Furthermore, this layer should adapt the wireless nature of the service to be provided to the connection-oriented nature of the services provided by protocols residing in the lower layers. The key functions to be performed in this layer are the following:

• Routing

Individual PDUs are routed through the network, based on the address information contained in the PDU, and network information available in the network nodes.

• Connection Management

In order to adapt the wireless service to be provided to the connectionoriented nature of lower layer protocols, a function is needed that initiates connection establishment, modification, and release. This connection management function is responsible for connections between network protocol entities.

Thus, the connection management function manages a network of connections between network nodes, called a wireless overlay network. Managing this wireless overlay network includes establishing connections and negotiating a suitable QoS and suitable traffic parameters. Furthermore, it includes modifying the traffic parameters of a connection when needed, or releasing the connection when it is no longer needed. For this purpose the connection management function requests the signalling system to perform the proper actions.

• Address Validation

In the BCL Network, the source address specified by the sending user in the PDU is checked against the source addresses that are allowed for the AAL connection on which the PDU arrives.

• Access Class Enforcement

The BCL Network enforces an end-system to comply with certain parameters (the access class) regarding the stream of packets that is sent into the network.

The parameters are agreed upon at the time of subscription.

• Address Screening

In order to allow for the creation of closed user groups, the destination address of PDUs is checked against a list of allowed addresses for the specified source (destination address screening). Furthermore, the source address of PDUs is checked against a list of allowed addresses for the specified destination (source address screening).

Protocols that reside in the BCL Network Layer are the BCL Network Access Protocol and the BCL Network Protocol.

ATM Adaptation Layer

Protocols for the ATM Adaptation Layer (AAL) provide for the transfer of AALSDUs of a variable but limited length along a connection from a source network layer entity to one or more destination network layer entities. The major function of these protocols is to adapt the variable length of AAL-SDUs to the fixed length of the ATM-SDUs.

The following key functions are performed:

• Segmentation and Reassembly

An AAL SDU can have any length (up to a certain maximum). It is transferred using a number of AAL PDUs with a size of 48 octets. Protocol Control Information

(PCI) is added during the segmentation, in order to enable correct reassembly at the receiver side.

• Error detection

Bit errors in the AAL PDU can cause malfunctioning of the reassembly function, or the routing function in the BCL Network Layer. As a result of this, data could unintentionally be delivered to the wrong end users terminal. From a security point of view, this is a very undesirable situation. Therefore, detection of errors will be performed in the AAL.

ATM Layer

The ATM Layer contains protocols that provide for the transfer of fixed size ATMSDUs between AAL entities along a previously established connection. In order to do this, the ATM-SDUs are transferred in ATM-PDUs (cells) from node to node using the Physical Layer service.

The following key functions are performed by the ATM layer:

• Identification of cells

Cells transferred using the Physical Layer service need to be identified, in order to know the virtual connection they belong to.

• Relaying of cells

In all intermediate ATM nodes, incoming cells are relayed to a certain outgoing link, based on the virtual connection they belong to.

Physical Layer

The Physical Layer provides for the transfer of cells between ATM nodes. The key function of this layer is:

• Transmission of cells

The ATM cells are transformed to a physical representation by the transmitting node. After propagation through the medium, the receiving node transforms the physical signals again to the original cell.

2.3.7 Relation to other Models and Architectures

Network architecture like the one given here, is not available in the literature. Some models and architectures can be found however. The ITU has defined a reference configuration for providing wireless services ([80]). It locates so called ATM switched capabilities and wireless service functions relative to the B-ISDN. These can be seen as functions of the physical and ATM layer and functions of the AAL and BCL network layer respectively. This reference configuration further identifies the interfaces between the real systems that are needed to provide the service. Also in [80], the ITU gives a protocol architecture that puts the protocols to be used at the UNI in relation with each other.

The BCLS protocol reference model is different from the B-ISDN protocol reference model defined in [13]. The latter considers the transfer of user information and the control of connections as conceptually different functions, i.e., it distinguished between a user and a control plane, while it is useful that they are initially looked at from an integrated perspective, describing the required overall behavior of the system under design. The B-ISDN protocol reference model only considers layers up to the AAL. The other layers are referred to as higher layers.

The control functions residing in the ATM, AAL, and network layer of our model would reside in the higher layers of the control plane. The other functions of the network layer reside in the higher layer of the user plane.

The SMDS specification (In the Introduction) does not provide protocol reference models or architecture as presented here.

2.4 AAL Protocols

In Section 2.3.6, segmentation, reassembly, and error detection have been identified as the key functions to be performed in the ATM Adaptation Layer. An AAL protocol should provide these functions. It should adapt the service provided by the ATM service over an ATM connection to a service that is more suitable for the transfer of BCL-PDUs.

Within the ITU, a number of AAL protocols have been identified to adapt the ATM service to a service more suitable for specific types of applications. Two of them can be used to support data communications. These are referred to as AAL 3/4 and AAL 5. We do not come up with a new proposal for an AAL protocol. We assume the use of one of the two protocols recommended by the ITU. For completeness we describe both the AAL 3/4 and the AAL 5 proposal in this section.

First in Section 2.4.1 we will elaborate on the way AAL protocols have been defined within the ITU. In Section 2.4.2 we will describe the service that is provided by the AAL protocols. We will describe the two candidate AAL protocols, AAL 3/4 and AAL 5 in Section 2.4.3 and Section 2.4.4 respectively. We will end this section with a comparison of both protocols in Section 2.4.5.

2.4.1 The ITU View of the AAL

The ITU describes the AAL in its Recommendations I.362 and I.363 ([78], [79]). Five types of AAL protocols have been identified. AAL type 1 has been defined to support constant bit rate services over an ATM network. AAL type 2 has been defined for variable bit rate services that need a constant delay, i.e., that need to transfer timing

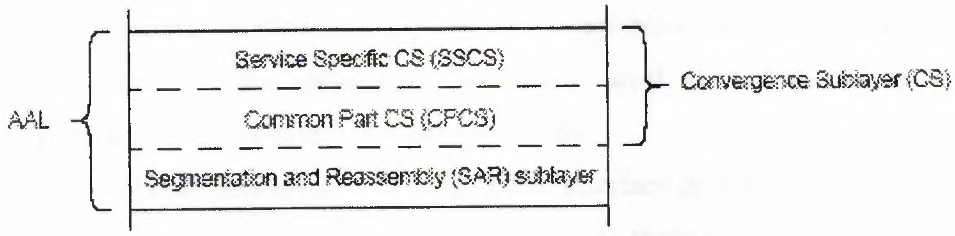


Figure 2-18: Sublayering of the ATM Adaptation Layer

information between source and destination. AAL types 3 and 4 are identical and are referred to as AAL 3/4. This protocol has been defined to support variable bit rate services that do not wish to transfer timing information, e.g., data services. A new AAL protocol, AAL 5 ([16]), has been defined, which also supports variable bit rate (data) services. This protocol is supposed to reduce protocol processing and transmission overhead compared to AAL 3/4.

Recommendation I.362 specifies the use of AAL 3/4 for the support of wireless services, but it is also stated that the standardization of other AALs for wireless services is for further study. From now on we will focus on AAL 3/4 and AAL 5, since these are the protocols that are best suited to support the BCLS.

The AAL has been sublayered in a Segmentation and Reassembly (SAR) sublayer and a convergence sublayer (CS), which has in turn been sublayered into a common part (CPCS), and a service specific part (SSCS) (Figure 2-18). The ITU specifies that for the support of wireless services, the SSCS will be empty.

For the AAL, only the user plane functionality is considered. Since the AAL is used on an end-to-end basis over an ATM connection, only very little control functionality is needed. An AAL connection corresponds always directly to an ATM connection, although multiplexing of different AAL connections onto a single ATM connection is possible in AAL 3/4.

Since the SSCS will be empty for the support of wireless services, the service provided by the AAL equals the service provided by the CPCS. In Recommendation I.363, the ITU has defined primitives for the service of the CPCS.

However, these primitives form already a refinement of the service primitives that are needed to describe the end-to-end behavior of the service. The primitives in the recommendation already describe how the local interface could be implemented. It refines the service primitives into a number of interactions at the local interface. At first, it suffices to describe the service at the highest level of abstraction, i.e., without describing how the service primitives can be refined.

Later on, we will elaborate on the refinement into interface primitives. The AAL 3/4 and the AAL 5 protocols provide almost the same service to their users. The AAL 5 proposals specify a few additional parameters for the service primitives.

However, these will not be used by the BCL network layer protocol. In the description of the AAL service, we omit these extra parameters. We describe the service as far as it is required by the BCL NL protocols.

2.4.2 AAL service

The AAL service is a connection-oriented service, which allows for the transfer of AAL-SDUs of any size up to 65 535 octets. The ATM-SDUs can be transferred along a previously established connection between a single source AAL-SAP to one or more destination AAL-SAPs. Like for the ATM service, establishment and release of a connection can be done either on demand or (semi-)permanently using management procedures.

Two modes of operation are identified: assured operations, and non-assured operations. In case of assured operations, every AAL-SDU is delivered with exactly the data content that the user sent (with a very high probability).

In case of non-assured operations, AAL-SDUs may be lost or corrupted with a certain probability. As an option, corrupted SDUs may be delivered to the receiving user with a notification that the SDU is corrupted in non-assured mode of operation (corrupted data delivery option). For use in a wireless environment, the ITU recommends an AAL that supports only the non-assured mode of operation.

Furthermore, it recommends not using the corrupted data delivery option. The AAL specification is not concerned with the connection establishment phase or connection release phase of the communications. This is considered to be handled in the control plane. In this chapter, we will also limit our scope to the data transfer phase of the communication.

In the data transfer phase, user data can be transferred with the following service primitives:

• AAL-DataRequest, with parameter
AAL-SDU; and

• AAL-DataIndication, with parameter
AAL-SDU.

If an AAL connection is available, an AAL-DataRequest primitive at a source CEP in an AAL-SAP will result in an AAL-DataIndication primitive at the destination CEP associated with the source CEP. In case multiple CEPs are associated with the source CEP, the AAL-DataRequest will result in AAL-DataIndication primitives at all destination CEPs. The AAL-SDU parameter of the indication primitives equals that of the corresponding request primitive(s).

Refinement of Service Primitives

In Recommendation I.363 ([78]) the ITU does not specify the service primitives, as we have specified them here. Instead, two possible refinements of the service primitives are specified. The service primitives are refined into a number of interactions, which we will call interface primitives. The SDU is exchanged across the interface (the implementation of a SAP) in Interface Data Units (IDUs), which are parameters of the interface primitives. The refinement applies to the interactions at a particular local interface, i.e., the implementer of an AAL protocol entity is free in choosing the refinement for that particular entity, as long as the implementation of the higher layer protocol entity uses the same refinement of the service primitives at the interface between the two entities.

The refinements of the AAL service primitives are called message mode, and streaming mode. In the message mode, the SDU of a service primitive (AALDataRequest or AAL-DataIndication) is exchanged as a single IDU in a single interface primitive. In the streaming mode the SDU is exchanged in a number of interface primitives, each exchanging a part of the SDU in its IDU.

Message mode, in the message mode, there is a one-to-one mapping from service primitives to interface primitives. Let us use the ITU terminology for the interface primitives, i.e., we call them invoke and signal, instead of the conventional request and indication for service primitives. In the message mode, the following interface primitives have been identified:

• AAL-UNITDATA-invoke, with parameter

AAL-IDU

This interface primitive implements the AAL-DataRequest primitive. The only parameter of the primitive is the IDU, with a maximum length of 65 535 octets.

• AAL-UNITDATA-signal, with parameter

AAL-IDU

This interface primitive implements the AAL-DataIndication primitive. It has as its parameter the IDU, which implements the SDU parameter.

Streaming Mode, in the streaming mode the service primitives are refined into series of interface primitives. The AAL-SDU parameter of the service primitive is exchanged in a series of consecutive AAL-IDU parameters of the interface primitives.

Extra interface primitive parameters are needed to identify to which SDU an IDU belongs. Furthermore, extra primitives are defined to enable user or service provider to abort the exchange of IDUs for a particular SDU at a certain interface.

The following interface primitives can be identified:

• AAL-UNITDATA-invoke, with parameters

AAL-IDU, More, and Maximum Length

This interface primitive is used to exchange a part of the SDU across the interface, in the form of the IDU parameter. A series of AAL-UNITDATA-invoke primitives implements a single AAL-DataRequest primitive. Apart from the IDU, the AAL-UNITDATA-invoke primitive has a More parameter that specifies whether the IDU contains the end of an AAL-SDU or not. Furthermore, the interface primitive exchanging the first IDU related to an SDU contains a Maximum length parameter, indicating an upperbound to the length of the AALSDU that is being exchanged by means of IDUs.

• AAL-UNITDATA-signal, with parameters
AAL-IDU, More, and Maximum Length
This interface primitive is used to exchange a part of the SDU across the interface, in the form of the IDU parameter. A series of AAL-UNITDATA-signal primitives implements a single AAL-DataIndication primitive. The More parameter is used in the same way as for the AAL-UNITDATA-invoke primitive.
Only the AAL-UNITDATA-signal primitive exchanging the first IDU related to an SDU contains a Maximum Length parameter, of which the meaning is the same as for the AAL-UNITDATA-invoke primitive.

• AAL-U-Abort-invoke
This primitive is issued across the local interface if a user that has already exchanged a number of AAL-IDUs related to a certain AAL-SDU wants to abort the exchange of the SDU.

• AAL-U-Abort-signal
This primitive is issued to indicate that the remote user has aborted the transfer of the AAL-SDU being received.

• AAL-P-Abort-signal
This primitive is issued to indicate that the provider is not able to complete the transfer of the AAL-SDU being received. Its use is initiated by the provider.

2.4.3 AAL 3/4

The AAL 3/4 protocol has been specified in ITU Recommendation I.363 ([76]) to provide the service specified in the previous subsection. Its general operation is as follows. Upon receipt of an AAL-DataRequest primitive a CPCS-PDU is constructed, containing the AAL-SDU as payload. This CPCS-PDU is segmented into segments of 44 octets. After that, a series of SAR-PDUs is constructed, each containing one of the segments as payload. These SAR-PDUs are sent to the destination AAL protocol entity as the payload of ATM cells. This entity strips the SAR-PDUs, and reassembles the CPCS-PDU using the obtained segments.

Finally, it extracts the AAL-SDU from the CPCS-PDU, and delivers it to the service user with an AAL-DataIndication primitive.

CPCS

Functions performed by the CPCS sublayer are:

- padding of the AAL-SDU to a multiple of 4 octets,

- delimiting of the AAL-SDU by means of a length field in the CPCS-PDU, and

- Limited error detection by means of the length field, and by means of (identical) tags in both the header and the trailer of the CPCS-PDU.

Figure 2-19 shows the format of the PDU passed by the CPCS to the SAR sublayer of the AAL 3/4. The following fields can be identified:

- Common Part Indicator (CPI)

This field is intended to be used for management purposes. Identification of management messages and indication of the counting units for the BASize and Length fields have already been foreseen.

- Begin Tag (BTag) and End Tag (ETag)

These tags associate the header and the trailer of a CPCS-PDU. The sender inserts the same value in the BTag and the ETag field of a given PDU and changes the value for each successive PDU. For a given PDU, the receiver checks the BTag with the ETag upon receipt, so that the loss or insertion of ATM cells containing a CPCS-PDU header or trailer can be detected.

• Buffer Allocation Size (BASize)

By means of this field, the sender informs the receiver of the maximum buffer requirements for the receipt of the CPCS-PDU. The BASize is equal to or larger than the length of the Payload field.

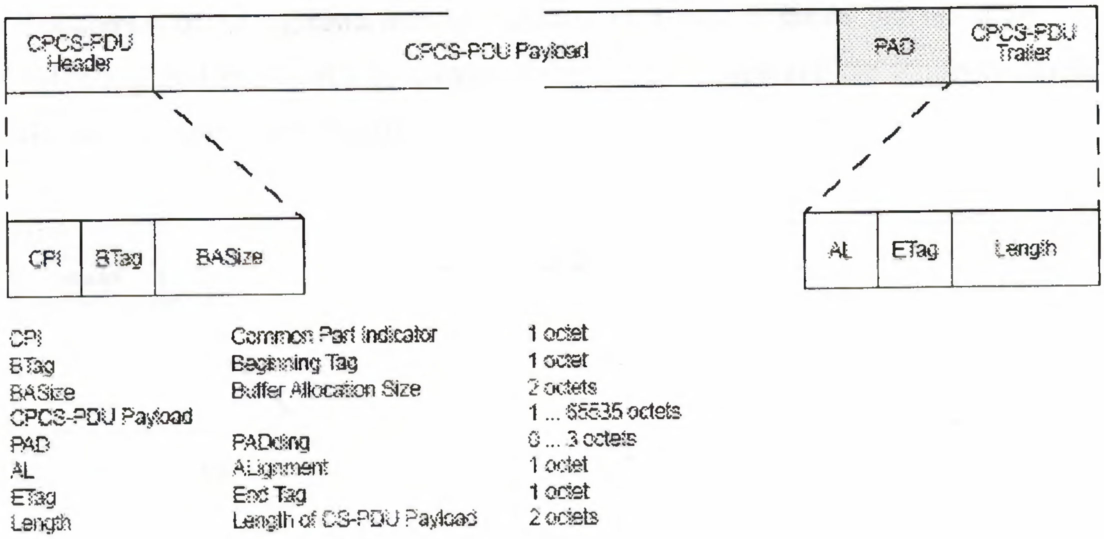


Figure 2-19: CPCS-PDU format for AAL 3/4

• CPCS-PDU Payload

This field contains the AAL-SDU.

• Padding (PAD)

This field complements the CPCS-PDU Payload field to a multiple of 4 octets.

• Alignment (AL)

This field is used to complement the length of the CPCS-PDU Trailer to 4 octets.

• Length

This field is used by the sender to inform the receiver of the length of the CPCS-PDU Payload field. The receiver uses the field to detect loss or gain of information.

SAR

Functions performed by the SAR are:

• Segmentation of CPCS-PDUs into 44 octet segments in the source protocol entity,

• Reassembly of the segments into the original CPCS-PDU in the destination entity,

• delimiting of CPCS-PDUs by means of the sequence type and the sequence number field, and by means of the length indication,

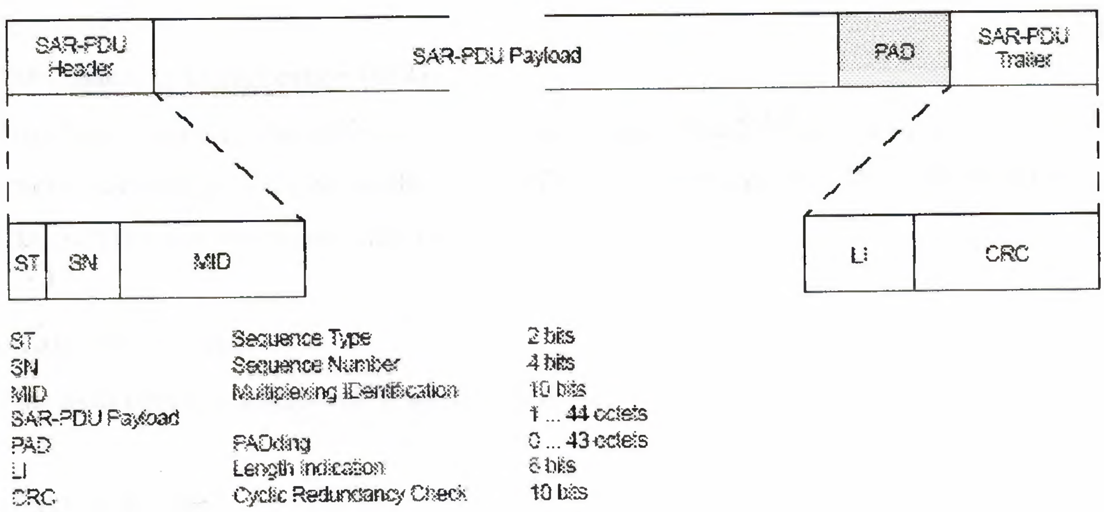


Figure 2-20: SAR-PDU format for AAL 3/4

• Error detection by means of the sequence type, sequence number and the CRC field, and

• Multiplexing and demultiplexing of different AAL connections by means of the MID field.

Figure 2-20 shows the format of the SAR-PDUs that are passed to the ATM layer.

The following fields have been defined:

• Sequence Type (ST)

This field identifies the SAR-PDU Payload as being either the first (Beginning Of Message, BOM), a middle (Continuation Of Message, COM), or the last (End Of Message, EOM) segment of a CPCS-PDU, or containing a complete CPCS-PDU (Single Segment Message, SSM).

• Sequence Number (SN)

The source assigns successive numbers (modulo 16) to this field, for the transfer of successive segments of the same CPCS-PDU. These SNs can be checked by the destination in order to detect missequencing, loss, duplication, or misinsertion of segments.

• Multiplexing Identification (MID)

This field is used to identify SAR-PDUs in case that either SAR-PDUs for different AAL connections are transferred on the same ATM connection, or if SAR-PDUs for different CPCS-PDUs are transferred interleaved.

• SAR-PDU Payload

This field contains a segment of the CPCS-PDU.

• Padding (PAD)

If the last or only segment of a CPCS-PDU has a length smaller than 44 octets, this field is used so that the total length of the segment and the PAD field is 44 octets.3

• Length Indication (LI)

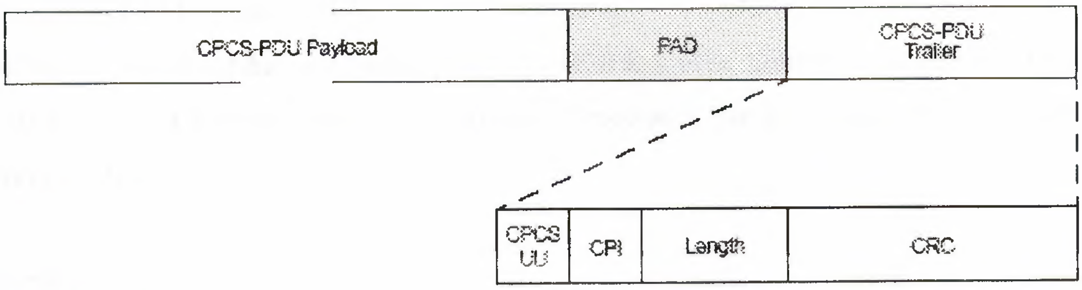
This field indicates the length of the SAR-PDU Payload field in octets.

• Cyclic Redundancy Check (CRC)

This field contains an error detecting code, calculated over the entire SARPDU. Special values of the fields have been defined for the case where the sending user wants to abort the transfer of a CPCS-PDU, e.g., because the AAL-U-Abort invoke interface primitive has been issued in streaming mode. This so-called Abort-SAR-PDU has its Segment Type set to EOM, and its Length field to 63. The Payload field may be set to 0, and is ignored by the receiver, just like the contents of previously received segments of the CPCS-PDU, currently being transferred.

2.4.4 AAL 5

The AAL 5 protocol has been proposed to alleviate the supposed overhead of the AAL 3/4 protocol ([16]). Almost all functionality of the protocol resides in the CPCS sublayer. The only function of the SAR sublayer is the segmentation and reassembly of messages. The SAR does not use PCI to perform this function, instead it uses the ATM User-to-User Information parameter of the ATM-DataRequest and -Indication primitives to detect the last segment of a message.4 The general operation of the protocol is as follows. Upon receipt of an AAL-DataRequest primitive, a CPCS-PDU is constructed, containing the AAL-SDU as payload. This PDU is segmented into 48 octet segments that are transferred to the destination AAL protocol entity in ATM cells. For the last segment of the PDU, the ATM User-to-User Information parameter is set. This parameter is transferred transparently with the ATM cell to the destination, which is now able to complete the reassembly of the CPCS-PDU. The receiver can extract the original AAL-SDU from the CPCS-PDU, and deliver it to the service user by means of an AALDataIndication primitive.



CS-PDU Payload		1 ... 65535 octets
PAD	PADding	0 ... 47 octets
CPCS-UU	CPCS User-to-User indication	1 octet
CPI	Common Part Indicator	1 octet
Length	Length of CS-PDU Payload	2 octets
CRC	Cyclic Redundancy Check	4 octets

Figure 2-21: CPCS-PDU format for AAL 5

CPCS

The functions carried out by the CPCS of the AAL 5 protocol are:

• Padding of the CPCS-PDU to a multiple of 48 octets,

• Delimiting of the AAL-SDU by means of a Length field in the CPCS-PDU trailer, and

• Error detection by means of the Length field and a CRC.

Figure 2-21 shows the format of the CPCS-PDU, which is passed by the CPCS sublayer to the SAR sublayer for segmentation. The following fields can be identified:

• **CPCS-PDU Payload**

This field contains the AAL SDU.

• **Padding (PAD)**

The Padding field complements the CPCS-PDU Payload field, so that the total length of the CPCS-PDU is a multiple of 48 octets.

• **CPCS User-to-User indication (CPCS-UU)**

This field is intended to be transparently transferred between the users of the CPCS.

• Common Part Indicator (CPI)

The use of this field has not been specified yet. It has been defined to align the CPCS-PDU trailer to a 8 octet boundary. A possible function is the identification of PDUs for layer management.

• Length

The Length field is used to indicate the number of octets the CPCS-PDU Payload field consists of. It can be used by the receiving entity to identify the end of the payload, and to detect any loss or gain of information (e.g., because of loss or misinsertion of segments).

• Cyclic Redundancy Check (CRC)

This field contains an error detecting code, calculated over the entire CPCSPDU.
In order to abort the transfer of a partially transmitted AAL-SDU, e.g., in case the AAL-U-Abort-invoke interface primitive has been issued, the Length field is set to 0.

SAR

The SAR-PDU of the AAL 5 does not contain any PCI. It only contains one 48 octet segment of the CPCS-PDU as its payload (Figure 2-22). The SAR performs the segmentation and reassembly functions. The transfer of the last segment of a CPCS-PDU is signaled by the ATM User-to-User indication parameter.

Figure 2-22: SAR-PDU format for AAL 5

2.4.5 Comparison

Both AAL 3/4 and AAL 5 provide a service that can be used by BCL network layer protocols, and both make use of the service provided by ATM. However, there are quite a number of differences between the two protocols. These are differences in the type of PDUs functions operate on, the functions performed, and the amount of PCI that is generated.

Type of PDUs functions operate on

Protocols for the AAL differ in the way the functions are performed. Functions can either be performed per CPCS-PDU (per packet) or per SAR-PDU (per segment). Table 2-3 classifies the AAL 3/4 and AAL 5 proposals according to the type of PDUs functions operate on. The segmentation and reassembly functions are not listed in this table, because they do not operate on either segments or packets. They provide for the transformation between segments, which form the payload of ATM cells, and packets.

In general, PCI needs to be exchanged by the communicating protocol entities, in order to be able to carry out a function. If a function is performed on a per segment basis, PCI will also be added to each segment, i.e., to the SAR-PDU. For a function, performed per packet, PCI will be added to the total packet, i.e., to the CPCS-PDU. It will be placed in the header or trailer, and thus be transferred with the first or last segment of the packet. Therefore, performing a function on a per segment basis tends to result in a larger overhead than performing a function on a per packet basis.
This seems to lead to the conclusion that all functions which need to exchange PCI should be carried out per packet (as it is done in the AAL 5 proposal). However, there are arguments against this conclusion. If the AAL is used in a BCL Network, user data proceeds from source to destination along a number of CLSs interconnected by ATM connections. In order to avoid the need for reassembly and segmentation in each CLS, it seems advantageous to perform the functions that need to be carried out in a CLS on a per segment basis.

This issue will be further discussed in Section 3.3. Furthermore, if error detection is performed per segment, the loss or corruption of segments can be detected earlier than if it is performed per packet. In the latter case, loss or corruption is only detected upon receipt of the last segment. For a CLS, this could imply that all other segments of the packet have already been forwarded to the next CLS. If error detection is performed per segment, forwarding of the segments after the corrupted or lost one can be avoided, which is more efficient.

As it can be observed from Table 2-3, AAL 3/4 performs a lot of functions per segment, while AAL 5 performs most functions per packet. This is reflected in the distribution of functionality over the SAR and CPCS sublayers. In principle, the SAR sublayer performs per-segment functions. The CPCS sublayer performs perpacket functions.

The AAL 3/4 has most of its functionality in the SAR sublayer, while the AAL 5 has an almost empty SAR sublayer, performing most functions in the CS sublayer.

Functions

There is one significant difference in the functions performed by AAL 3/4 and AAL 5. The AAL 3/4 protocol performs a multiplexing function, which is not performed by the

Table 2-3: Classification of AAL proposals

function	AAL 3/4	AAL 5
padding	per segment + per packet	per packet
AAL-SDU delimiting	per segment + per packet	per packet
error detection	per segment + per packet	per packet
multiplexing	per segment	not provided
abort	per segment	per packet

AAL 5 protocol. In a wireless environment, a multiplexing function enables a protocol entity to transfer several AAL-SDUs simultaneously. The SAR-PDUs carrying the SDUs are transmitted interleaved on the outgoing ATM connection. This is especially useful if the interface to the protocol operates in streaming mode. Then, the AAL-IDUs constituting an SDU can be passed to the AAL interleaved with the IDUs of other SDUs. AAL 3/4 is able to perform multiplexing because it can identify SAR-PDUs by means of the MID field. The importance of this possibility will be discussed in Section 3.3.

Generated PCI

Figure 2-23 gives an example of the segmentation of an AAL-SDU by the AAL 3/4 protocol. First the CPCS header and trailer are added to the user data, next the obtained CPCS-PDU is segmented into 44-octet chunks. SAR headers and trailers are added to these segments, in order to enable reassembly, and detection of transmission errors. The constructed SAR-PDUs are passed to the ATM layer as ATM-SDUs. These are transmitted by the ATM layer, after adding an ATM header. Padding is used to complement the CPCS-PDU to a multiple of 4 octets, and to complement the last SAR-PDU to a 48-octet boundary.

The total number of octets, , passed to the ATM layer as ATM-SDUs, for the transfer of a single AAL-SDU with a length of octets is given by

$$L_{3/4} = 48 * [(N + 8) / 44] \qquad (3.1)$$

This can be explained as follows. Each ATM-SDU contains 48 octets. The number of ATM-SDUs to be transferred equals the number of 44-octet segments that are needed to carry the AAL-SDU plus 8 octets of header and trailer.

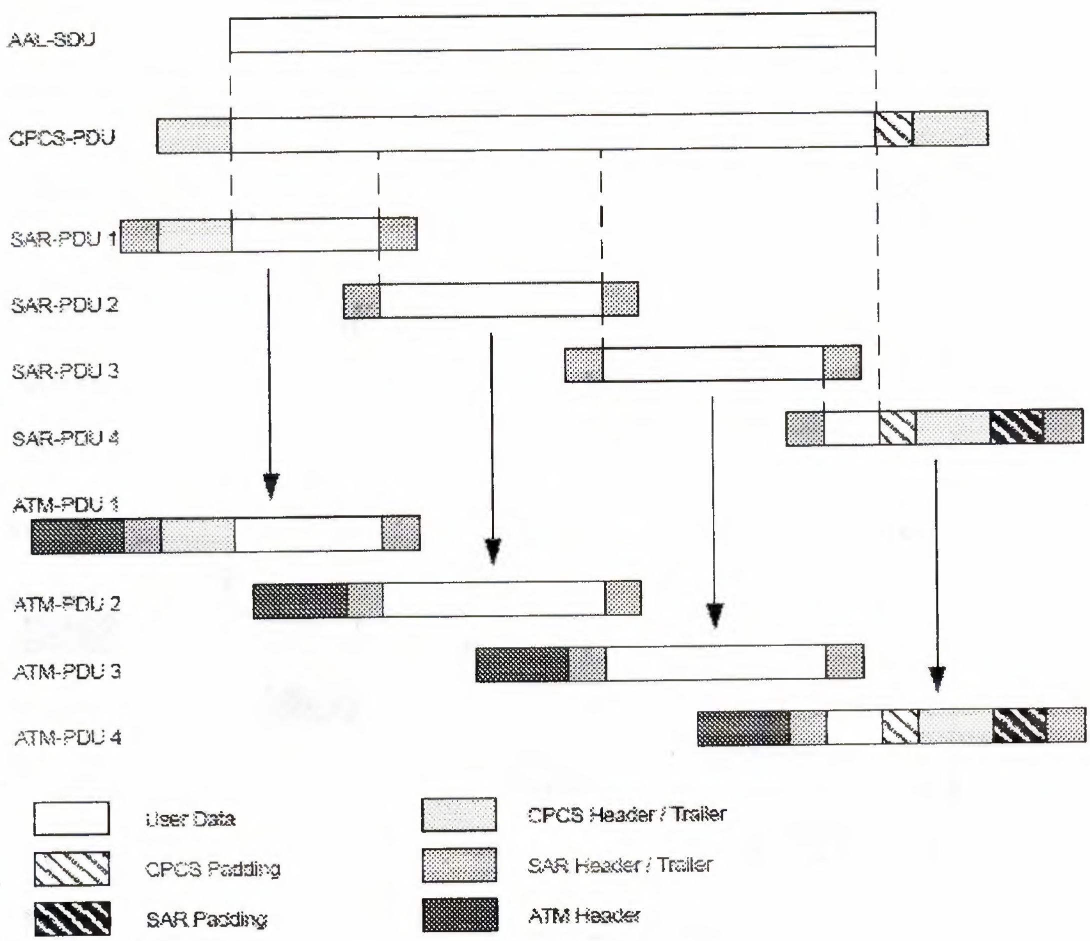


Figure 2-23: Segmentation in AAL 3/4

Figure 2-24 illustrates the segmentation of an AAL-SDU by the AAL 5 protocol. First, CPCS Padding and a CPCS trailer is added to the SDU, so that the resulting CPCS-PDU consists of a multiple of 48 octets. Next, the CPCS-PDU is segmented into 48 octet chunks, which are transferred as the payload of ATM cells.

For the AAL 5, the total number of octets passed to the ATM layer for the transfer of a single AAL-SDU, $\mathbf{L_5}$, is given by

$$\mathbf{L_{3/4} = 48 * [\,(N + 8\,)\,/\,48].} \qquad \mathbf{(3.2)}$$

Each ATM-SDU to be transferred contains 48 octets. The total number of octets is the product of 48 and the number of 48-octet segments needed to carry the AALSDU and the 8 octets of header and trailer information.

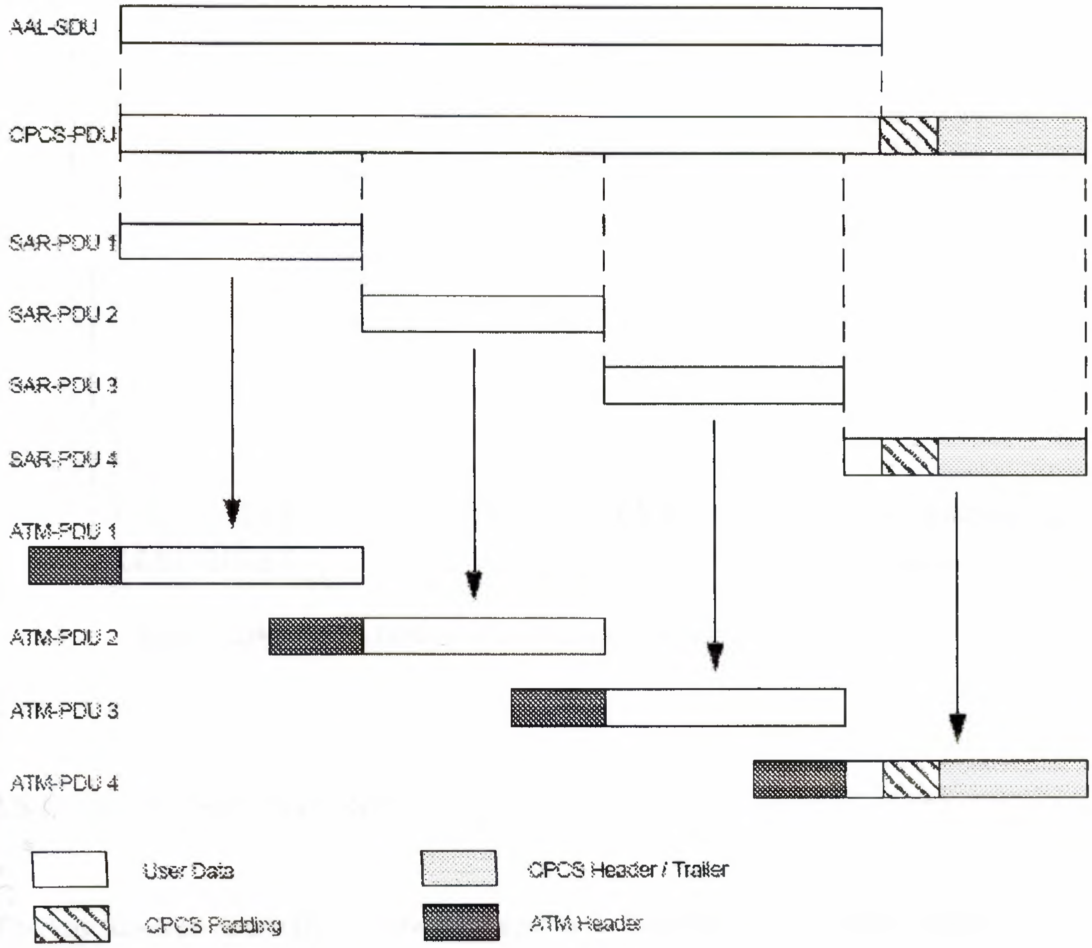


Figure 2-24: Segmentation in AAL 5

In order to show the difference in efficiency between the two AAL proposals, Figure 2-25 shows the relation between *N*, the AAL-SDU length, and **ΔL** , the proportional difference in efficiency between AAL 3/4 and AAL 5. This difference has been defined as follows:

$$\Delta L = 100\ (\ L_{3/4} - L_5\ )\ /\ L_{3/4}$$

For large *N*, **ΔL** converges to **100 × (1– 44 / 48).** This means that the AAL 5 protocol needs approximately 8.3% less bandwidth than the AAL 3/4 protocol for the transfer of large packets.

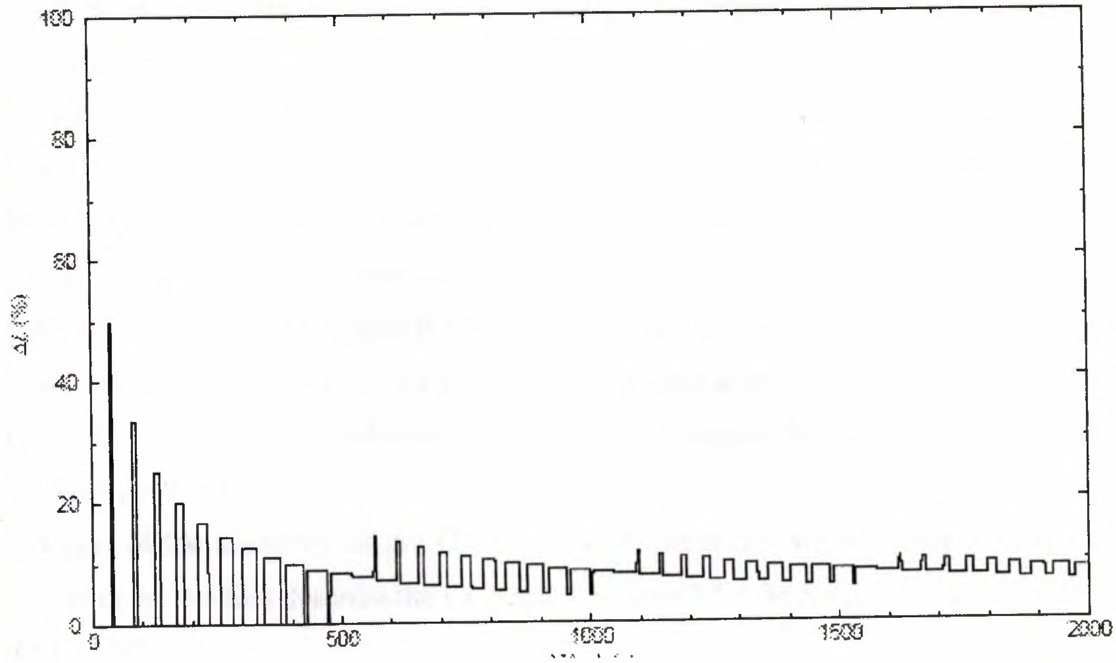


Figure 2-25: Relative Difference in Efficiency between AAL 3/4 and AAL 5

2.5 Network Layer Protocols

The key function of the BCL Network Layer is the routing of PDUs containing the users' BCL-SDUs from source to destination. Additional functions to be performed include address validation, address screening, access class enforcement, and connection management. The BCL Network Layer should perform the conversion from the wireless BCLS to the connection-oriented AAL service.

In the network architecture for the direct method, we have defined two types of network layer protocols, the BCL Network Access Protocol, and the BCL Network Protocol. The first one is intended for the access of an end-system to the BCL Network. It should take care of maintaining connectivity between the user and the network, address validation, address screening, access class enforcement, and delivery of a PDU containing the users BCL-SDU from the source to the proper access CLS, or from an access CLS to the proper destination. The latter should maintain the connectivity between CLSs in the network, and route the PDU from the CLS connected to the source to the CLS connected to the destination.

Network layer protocols have been defined in different communities. The ITU is currently defining two protocols for the provision of BCDS on B-ISDN ([80], [86]). The Wireless Network Access Protocol (CLNAP) is intended to be used between an end-system and a CLS. The Wireless Network Interface Protocol (CLNIP) should be used between CLSs. Similar protocols have been defined to provide SMDS over an ATM-based access network. These are called SMDS Interface Protocol for Wireless Service (SIP-CLS), and the Inter-exchange Carrier Interface Protocol for Wireless Service (ICIP-CLS) respectively.

Because of the similarity of the ITU and SMDS protocols, we will only describe the former ones. We first describe the CLNAP in Section 2.5.1. In Section 2.5.2 we describe the CLNIP.

2.5.1 CLNAP

The Wireless Network Access Protocol (CLNAP) is being defined in ITU Recommendation I.364 ([80], [86]). Its general operation is as follows. Upon receipt of a BCL-DataRequest, it constructs a CLNAP-PDU. An AAL connection to the proper access CLS in the BCL Network is selected and an AAL-DataRequest is issued at the proper CEP. In the CLS, upon receipt of the AAL-DataIndication containing the PDU, source address validation, access class enforcement, and destination address screening is performed, and the addresses are considered by the routing function. In the access CLS on the receiver side, the proper AAL connection to an end-system is selected and source address screening is performed. The CLNAP-PDU is transferred over the connection by issuing an AAL-DataRequest primitive at the corresponding CEP. Upon receipt of the PDU in the end-system as the SDU of an AAL-DataIndication primitive, the BCL-SDU is extracted from the PDU and delivered to the service user by means of a BCL DataIndication primitive.

Functions performed by the CLNAP are:

• Delimiting of the CLNAP-SDU,

• Addressing, in order to allow for the selection of the destination and indication of the source,

• Carrier selection, to allow the user to explicitly select a preferred connection on a per SDU basis,

• QoS selection, to allow for selection of the desired QoS on a per SDU basis,

•Padding of the SDU to a multiple of 4 octets, and optionally

• Error detection by means of a CRC.

Figure 2-26 shows the format of the PDU passed by the CLNAP to the AAL. The following fields can be identified:

• Destination Address

This field indicates to which protocol entity(ies) the CLNAP-PDU is destined.

• Source Address

This field indicates which protocol entity originated the CLNAP-PDU.

• Higher Layer Protocol Indicator (HLPI)

This field identifies to which protocol entity using the service of the CLNAP the SDU must be delivered.

• PAD Length

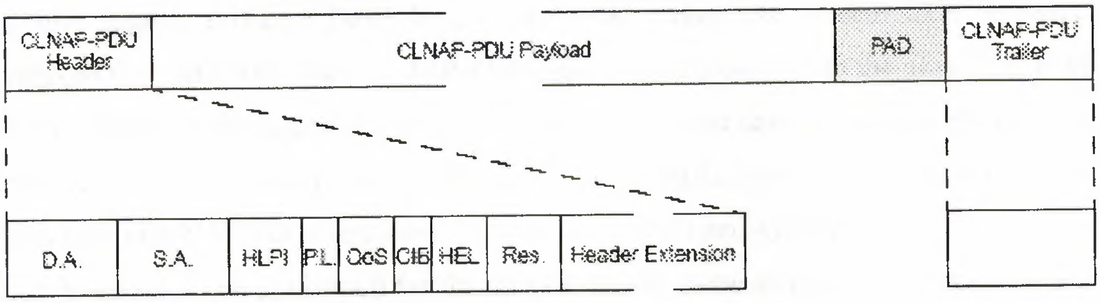
This field indicates the length of the PAD field in octets.

• Quality of Service (QoS)

This field indicates the QoS requested by the user for the transfer of the SDU.

• CRC Indication Bit (CIB)

This field indicates the presence of a CRC field



D.A.	Destination Address	4 octets
S.A.	Source Address	4 octets
HLPI	Higher Layer Protocol Identifier	6 bits
P.L.	PAD Length	2 bits
QoS	Quality of Service	4 bits
CIB	CRC Indication Bit	1 bit
HEL	Header Extension Length	3 bits
Res.	Reserved	2 octets
Header Extension		0 ... 20 octets
CLNAP-PDU Payload		1 ... 9188 octets
PAD	PADding	0 ... 3 octets
CRC	Cyclic Redundancy Check	4 octets (optional)

Figure 2-26: CLNAP-PDU format

- **Header Extension Length (HEL)**

This field indicates the length of the Header Extension field in 32-bit words.

- **Reserved**
- Header Extension

The use of this field has not been defined yet in [81]. It can for instance be used for the indication of the required carrier.

- **CLNAP-PDU Payload**

This field contains the BCL-SDU.

- **Padding (PAD)**

This field complements the CLNAP-PDU Payload to a multiple of 4 octets.

- **Cyclic Redundancy Check (CRC)**

This field is optional. If it is present it contains the result of a CRC32 calculation performed over the entire CLNAP-PDU.

2.5.2 CLNIP

The Wireless Network Interface Protocol (CLNIP) is also being defined in ITU Recommendation I.364 ([76], [81]). It is intended for use between CLSs. Its PDU format is only slightly different from the CLNAP PDU format. An Access CLS converts a received CLNAP-PDU into a CLNIP-PDU, and forwards this one to the next CLS, based on the address information of the PDU. Transit CLSs just have to forward the PDUs. In the Access CLS, connected to the destination of the PDU, the CLNIP-PDU is converted back into a CLNAP PDU, and forwarded to the relevant end-system.

The functions to be performed for the CLNIP are the same as those to be performed for the CLNAP, except for the carrier selection function, which is not performed. Furthermore, an additional function called encapsulation has been defined, which may be applied in Access CLSs.

Encapsulation is a technique where an entire CLNAP-PDU, including its header, is transferred in the payload field of a CLNIP-PDU. In the Access CLS, connected to the destination, the CLNAP-PDU is extracted again. This is called decapsulation.

If encapsulation is not applied, only the payload of the CLNAP-PDU is transferred by the CLNIP-PDU. Its header is replaced by a CLNIP-PDU header. These headers may be identical.

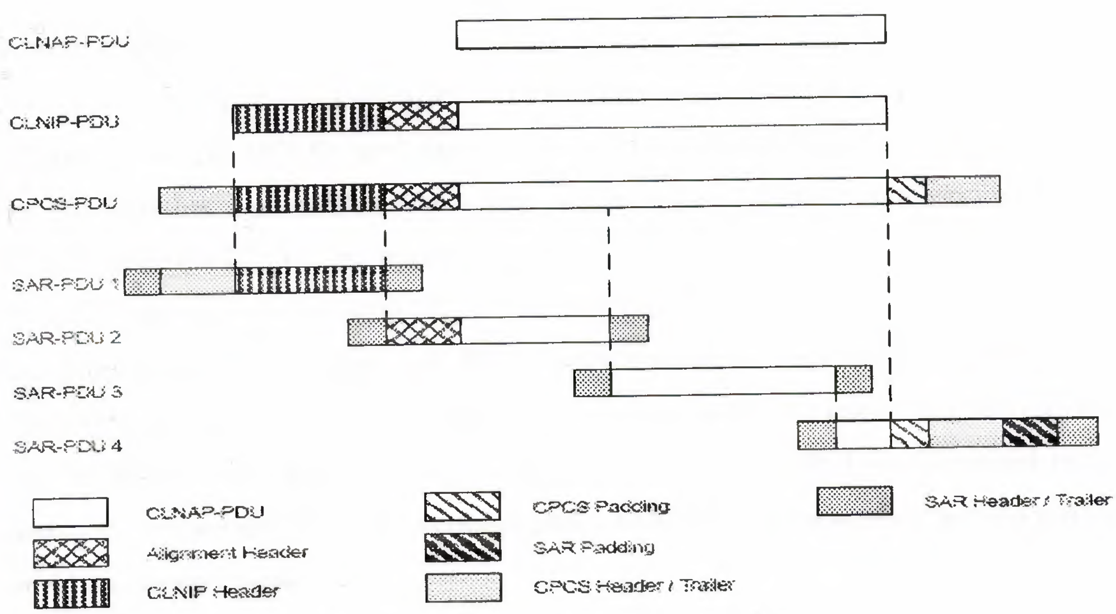


Figure 2-27: Encapsulation of a CLNAP-PDU

The PDU format of the CLNIP is identical to the one for the CLNAP (Figure 2-26) if encapsulation is not applied. If encapsulation is applied, the optional CRC field is not used. Furthermore, the header is extended with a Header Extension Post-Pad. This field pads the Header Extension field to a total of 20 octets, so that the total length of the CLNIP-PDU header is 40 octets. If AAL 3/4 is used to transfer the PDU, this header plus the CPCS header (4 octets) fit exactly in a single SARPDU Payload field (44 octets). Thus, in case of encapsulation an extra ATM cell, containing the new CLNIP-PDU header, is transferred before the cells containing the CLNAP-PDU are forwarded. An extra 4 octet alignment header is inserted before the CLNAP-PDU to fill the space that was originally occupied by the CPCS-PDU header, which is now positioned before the CLNIP-PDU (see Figure 2-27).

2.6 Summary and Concluding Remarks

In this chapter, we have presented an architectural framework for the provision of a wireless service with an ATM-based B-ISDN. It is obtained using a method of functional decomposition of a service provider into a number of protocol entities cooperating according to the rules of a protocol, and an underlying service, which they use for their communication.

As a starting point, abstract descriptions of the service to be provided, i.e., the Broadband Wireless Service (BCLS), and the service of the underlying system, i.e., the ATM service, have been given. These have been derived from available documents concerning SMDS, and from ITU recommendations.

We have introduced two candidate network architectures, in which different protocols and protocol entities are identified. The network architectures are defined by recursively decomposing the BCLS until the ATM service is obtained as the underlying service. The first architecture assumes that only functionality up to the ATM layer is present in the network. Conversion of the connectionoriented ATM service to a wireless service is done on an end-to-end basis.

This is called the indirect method of providing a wireless service. The second network architecture is based on the direct method. It assumes additional functionality in the network, which converts from the connection-oriented ATM to wireless communications. According to this architecture, special nodes, called Wireless Servers (CLSs) are installed in the network, to perform this conversion by routing packets from incoming to outgoing ATM connections, based on the address information contained in the packet. The latter network architecture is the most suitable one for situations where individual end-systems need to transfer packets to a lot of different destinations, since it reduces the number of needed ATM connections.

A protocol reference model has been derived from the network architectures. It identifies two layers on top of the ATM layer, which are needed to provide the BCLS. These are the ATM Adaptation Layer and the Broadband Wireless Network Layer.

Finally, we have overviewed protocols that can be used at the different layers, identified in the network architecture. These protocols are the AAL 3/4 protocol and the AAL 5 protocol for the AAL, and the CLNAP and the CLNIP for the BCL Network Layer. They are all being standardized within the ITU. It can be concluded that the installation of CLSs within the B-ISDN is the most important extension that is needed to let it provide a wireless service.

CHAPTER THREE

IMPLEMENTATION ASPECTS OF WIRELESS SERVERS

In this chapter, we focus on the direct method of providing a wireless service. We study the implementation of the CLSs, which are responsible for routing data packets through the ATM network. In the network architecture, presented in the previous chapter, we have identified the functional (protocol) entities that constitute a CLS. In this chapter, we study the problem how a real system can implement these functional entities. We present a number of candidate implementation architectures, which describe how the functionality of the entities to be implemented can be divided over modules of a CLS. These modules are the constituting parts of a real system, e.g., processor boards. For the purpose of designing implementation architecture from the network architecture, we transform the functional entities and their interactions to modules and their interconnection (see Figure 3-1).

We discuss the implementation of a CLS in more detail with respect to a number of important issues. These are the reassembly of network layer packets, the copying and the buffering of data within the CLS, and the integration of a CLS within an ATM switch. Furthermore, we analyse the proposed implementation architectures with respect to some of the design criteria given in Chapter 1, i.e. availability and scalability.

First, in Section 3.1, we present the various implementation architectures that can be used to implement a CLS. Then, in Section 3.2, we describe the functionality of the identified modules in more detail. In Section 3.3, we address a number of implementation issues that heavily influence the ultimate design, and have a strong impact on the performance of the system. Finally, in Section 3.4, the evaluation of the implementation architectures according to a number of criteria will be discussed.

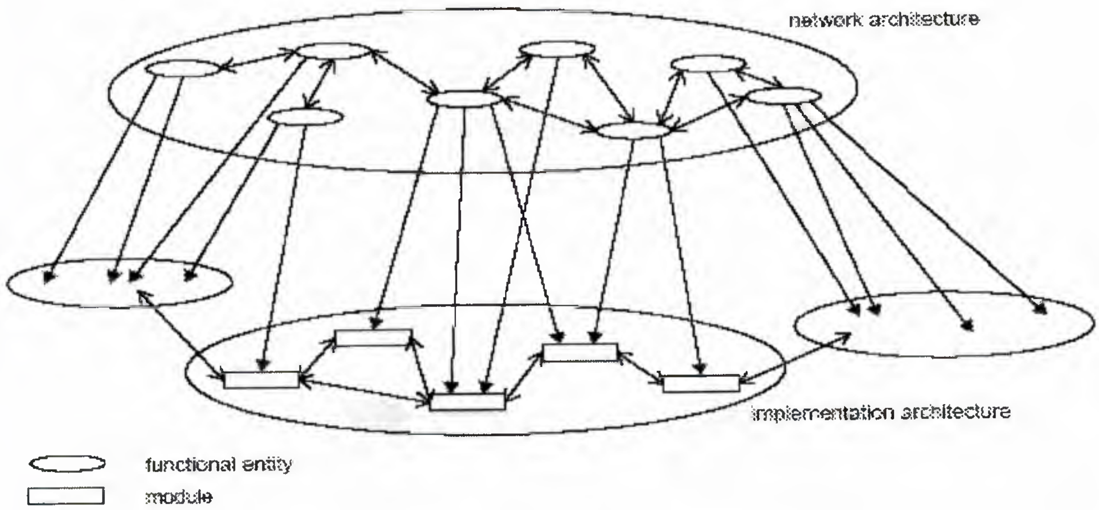


Figure 3-1: Design of Implementation Architecture

3.1 Implementation Architectures

Once the functionality of a CLS has been well specified, a first step towards its implementation is to design implementation architecture. In the implementation architecture, we determine which modules will constitute the functionality, the functionality of the individual modules, the interactions between the modules, and the way they are physically interconnected. In this section, we identify six different implementation architectures, starting from a basic one, and ending with a complex one aiming at high performance, which can easily be integrated in an ATM switch.

First, in Section 3.1.1, we discuss the implementation architecture of an ATM switch, and recall the functional entities that should be implemented in a CLS.
Next, in Sections 3.1.2 through 3.1.7, we present the six different CLS implementation architectures.

3.1.1 General

Before discussing the implementation of a CLS, let us spend some time on the implementation of an ATM switch. This is useful since a CLS and an ATM switch have much in common. Both should route ATM cells that arrive on an incoming link to the proper outgoing link, and update some of the protocol control information contained in the cells. Furthermore, one might want to integrate a CLS in an ATM switch for a number of reasons, which will be discussed in Section 3.3.4.

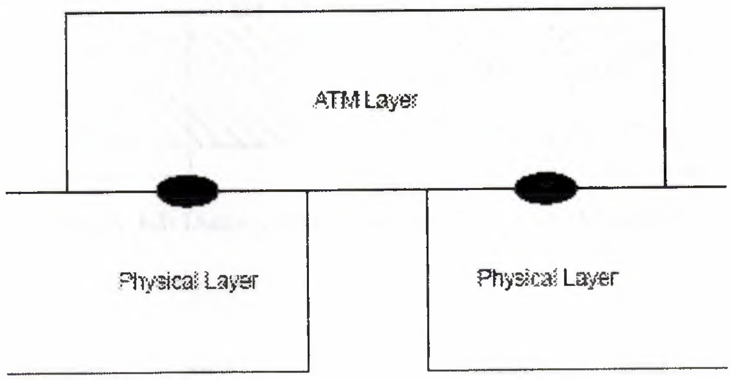


Figure 3-2: Functional Model of an ATM Switch

In Chapter 3, we have already identified an ATM switch (or node) as a real system that will be needed to provide the BCLS. Figure 3-2 shows a functional model of an ATM switch, which has been derived from part of the model in Figure 2-4. A number of protocol entities, performing functions residing in specific layers of the BCL Protocol Reference Model, can be identified in the functional model. Besides Physical Layer processing, the switch has to perform the switching function of the ATM layer.

In most implementations functionality is distributed over a number of modules (see Figure 3-3). Figure 3-3 describes the same functionality as Figure 3-2. However, whereas the latter shows an abstraction of the system, with boundaries and SAPs between abstract subsystems (i.e., protocol entities), the former shows the mapping of these abstract subsystems onto real subsystems, i.e., modules. In Figure 3-3, different modules are represented by different shadings.

Points of interaction between modules are denoted as open ovals. The order in which the different modules operate on an arriving data unit is indicated by an arrow.

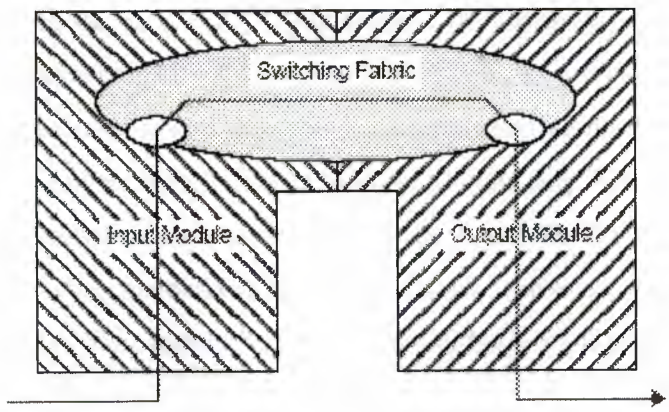


Figure 3-3: Distribution of Functionality in an ATM Switch

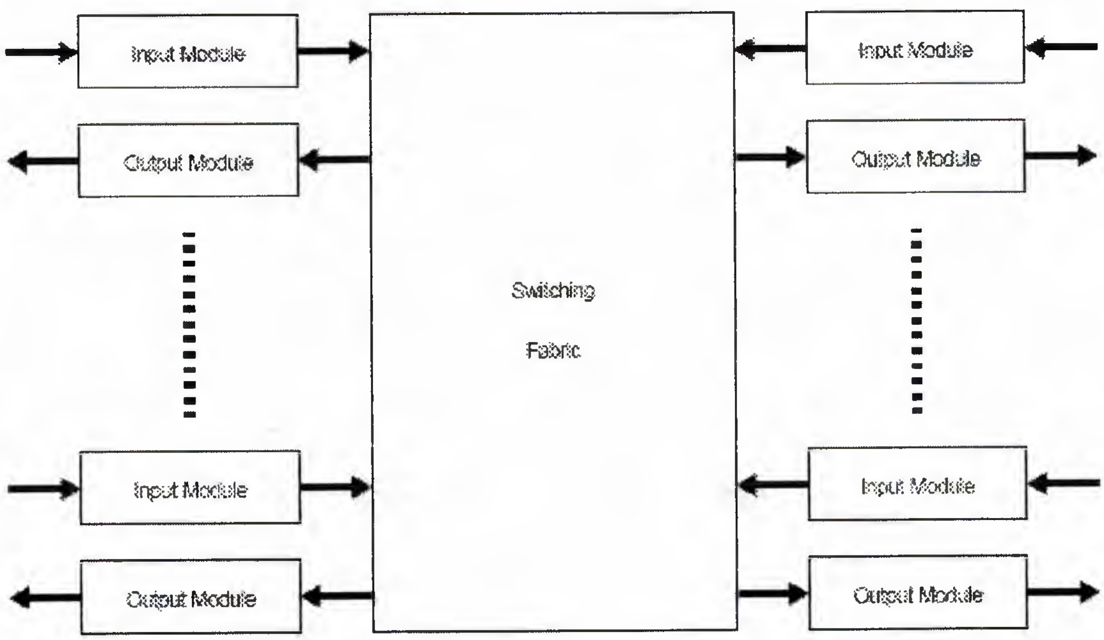


Figure 3-4: Architecture of an ATM Switch

Three types of modules can be identified in an ATM switch. The main module is the Switching Fabric. Its function is to forward ATM cells from the Input Modules to the proper Output Modules. The format of the switch data units, i.e., the unit of information the Switching Fabric operates on, can but does not necessarily have to be the same as the format of the ATM cell. The path a switch data unit has to take through the Switching Fabric is determined by a routing tag, i.e., a header field of the switch data unit, and information present in the fabric. An Input Module performs Physical Layer functions, and ATM header processing. It will convert ATM cells to switch data units, e.g., by segmentation. Furthermore, it will determine a routing tag for the switch data unit. An Output Module will convert switch data units to ATM cells, e.g., by removing the routing tag and possibly by reassembling a number of switch data units. It will perform ATM and Physical Layer processing, needed to submit the cell over the outgoing ATM interface. In general, it is possible to shift some of the functionality between Input Module, Switching Fabric and Output Module.

According to this distribution of functionality, a switch architecture as shown in Figure 3-4 is obtained. A large number of Input and Output Modules is connected to a single Switching Fabric.

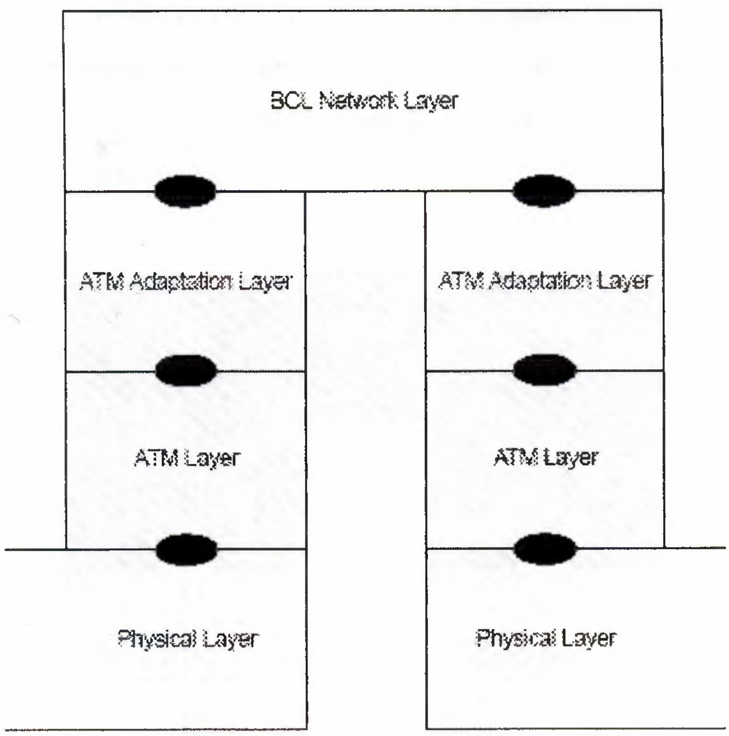


Figure 3-5: Functional Model of a CLS

A functional model of a CLS is shown in Figure 3-5. Switching is now performed at the BCL Network Layer. Furthermore, the CLS has to perform processing for the Physical Layer, the ATM Layer, the ATM Adaptation Layer, and the BCL Network Layer.

In general, a CLS will also have Input and Output Modules, which perform Physical Layer and ATM Layer functions, just like in an ATM switch.

These can be very similar to the ones described above for an ATM switch. Additionally, modules performing functions of the AAL and BCL Network Layer can be found in a CLS. These will be called Wireless Service Modules (CLSMs).

In general, a CLS connects to multiple input links; hence an Interconnection Structure (IS) is needed as well. This IS forwards data units between the different modules in the CLS. An example of such an IS an ATM Switching Fabric.

Additional to the modules that will be identified below, a CLS will contain one or more modules that perform management tasks. Functions of such a module are among others to download and update tables in other modules, and to detect failures of other modules. Such a management module will communicate with the other modules in the CLS either using the common IS, other via a dedicated interconnection structure. We do not take the management modules into further consideration

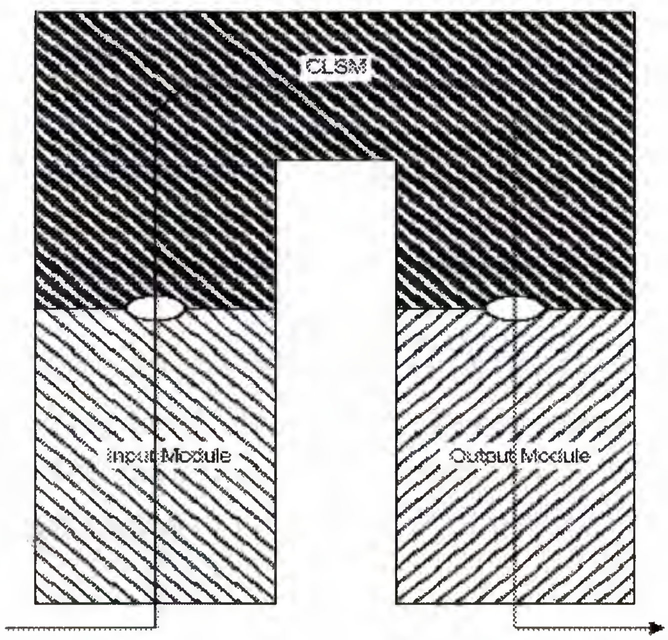


Figure 3-6: Distribution of Functionality (Architecture 1)

In the following subsections, a number of different implementation architectures will be presented. They differ in the modules that are identified, and in the way these modules are interconnected, i.e., the transformation from network architecture to implementation architecture has been performed differently.

3.1.2 Architecture 1

According to Architecture 1, a CLS consists of a single CLSM. No IS is used. The CLS terminates only a single ATM link (Figure 3-6). The Input and Output Modules for this link are connected directly to a CLSM. The CLSM performs all the processing of the AAL and BCL Network Layer.

ATM cells received by an Input Module are processed, and their payload and an internal connection identifier are forwarded to the CLSM. From the incoming cell stream, the CLSM (virtually1) reassembles the packet. A routing function, implemented in the CLSM determines the proper outgoing ATM connection, based on the address information contained in the packet. Next, the CLSM (virtually) segments the packet into cell payloads and forwards them to the Output Module together with an identifier for the

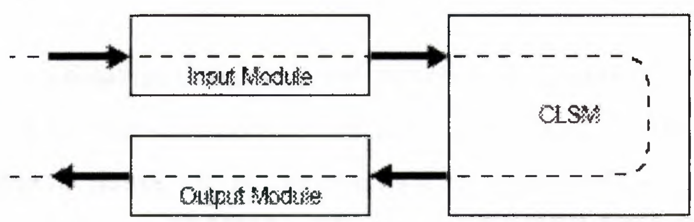


Figure 3-7: Implementation Architecture 1

required outgoing ATM connection. This module generates a cell header and transmits the ATM cell on the outgoing link. We obtain an architecture consisting of a single Input Module, a single CLSM, and a single Output Module (Figure 3-7). In this figure, the dashed line indicates the path of (the segment of) a BCL-PDU.

This implementation architecture is the most basic one that can be identified. The architecture can perform all functions that a CLS should perform. However, it allows only for small capacity CLSs, which are connected to only a single ATM link, since no switching is performed. We now identify a second alternative, that allows for CLSs of a larger capacity.

3.1.3 Architecture 2

According to Implementation Architecture 2, a CLS consists of several CLSMs interconnected by an IS. We distinguish between Input CLSMs, performing processing for incoming packets and routing, and Output CLSMs performing the processing for outgoing packets. Each Input or Output CLSM is connected to an Input or Output Module respectively (Figure 3-8).

ATM cells received by an Input Module are processed, and their payload is forwarded to the corresponding Input CLSM, together with an internal connection identifier. From the incoming cell stream, the Input CLSM (virtually) reassembles the packet. A routing function, implemented in the Input CLSM determines the proper outgoing ATM connection and link, based on the address information contained in the packet.

The packet and an indication of the required outgoing connection are forwarded via the IS to the Output CLSM that is responsible for the outgoing link. The packet is forwarded in a format suitable for the IS, i.e., in switch data units. This could be data units with the same length as a single segment. The Output CLSM (virtually) segments the packet into cells, and forwards them to the connected Output Module (possibly interleaved with other packets). This module generates a cell header and transmits the ATM cell on the outgoing link.

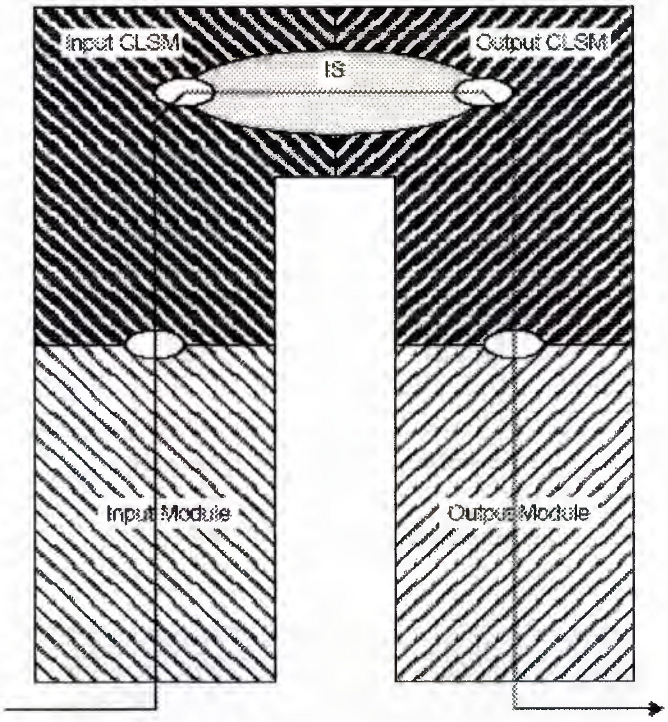


Figure 3-8: Distribution of Functionality (Architecture 2)

Using this distribution of functionality, implementation architecture is obtained with one Input Module, one Input CLSM, one Output CLSM and one Output Module per ATM link (Figure 3-9). The IS can be such that it switches segments or whole packets, depending on the operation of the CLSMs.

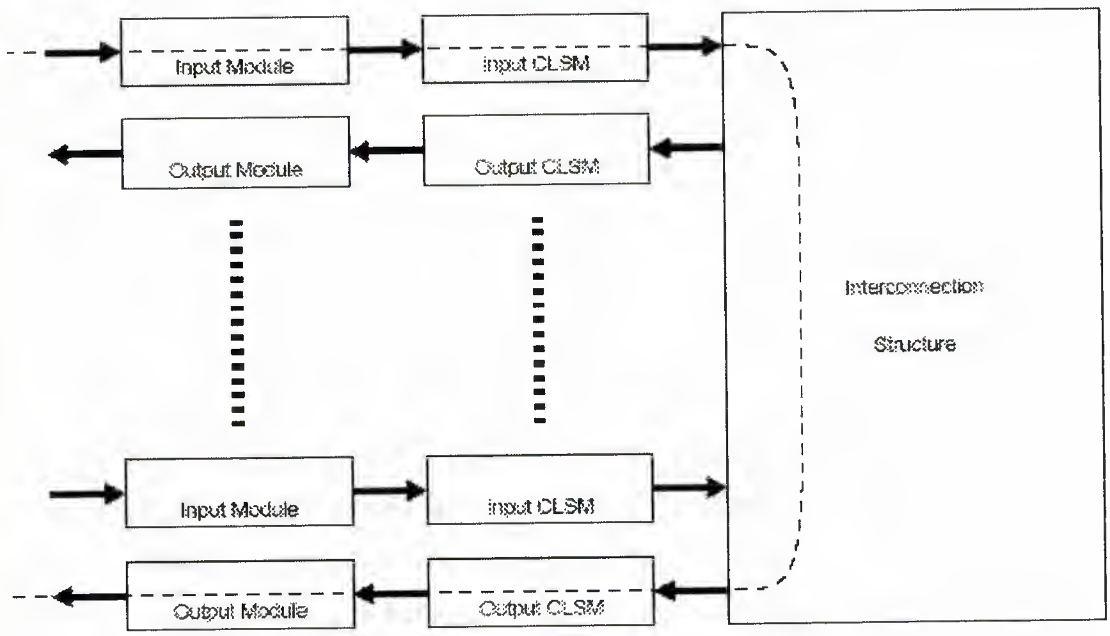


Figure 3-9: Implementation Architecture 2

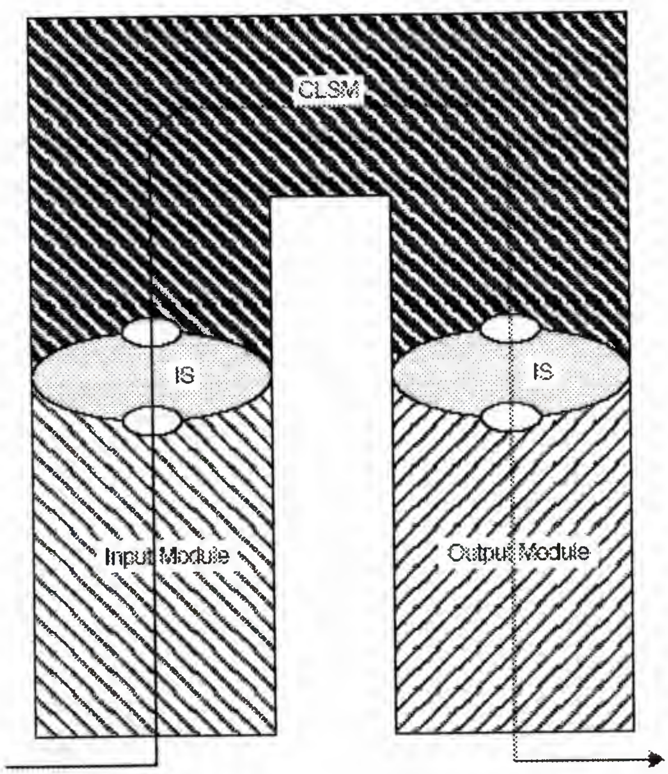


Figure 3-10: Distribution of Functionality (Architecture 3)

Using this architecture, a CLS can be scaled until the throughput of the IS is exceeded. Each ATM link has its own dedicated CLSM (Input as well as Output). A possible bottleneck is the throughput of a single CLSM, which restricts the throughput of a link. In order to relieve this problem, another implementation architecture is introduced, which uncouples the Input and Output Modules from the CLSMs, thus allowing for independent scalability of the CLSMs.

3.1.4 Architecture 3

According to Architecture 3, Input Modules, Output Modules, and CLSMs are all grouped around a central IS. No fixed relationship between an Input or Output Module and a CLSM exists. An Input Module forwards a cell to one of the CLSMs, depending on the ATM connection the cell belongs to, i.e., depending on the VPI/VCI (Figure 3-10). ATM cells received by an Input Module are processed and their payload and an internal connection identifier are forwarded via the IS to a CLSM. To which CLSM the cell is forwarded depends on the incoming ATM connection (VPC/VCC) on which the cell arrives. An ATM connection is terminated in a single CLSM, which maintains state information for that connection. This CLSM (virtually) reassembles a packet from the incoming cells. A routing function determines the proper outgoing link and

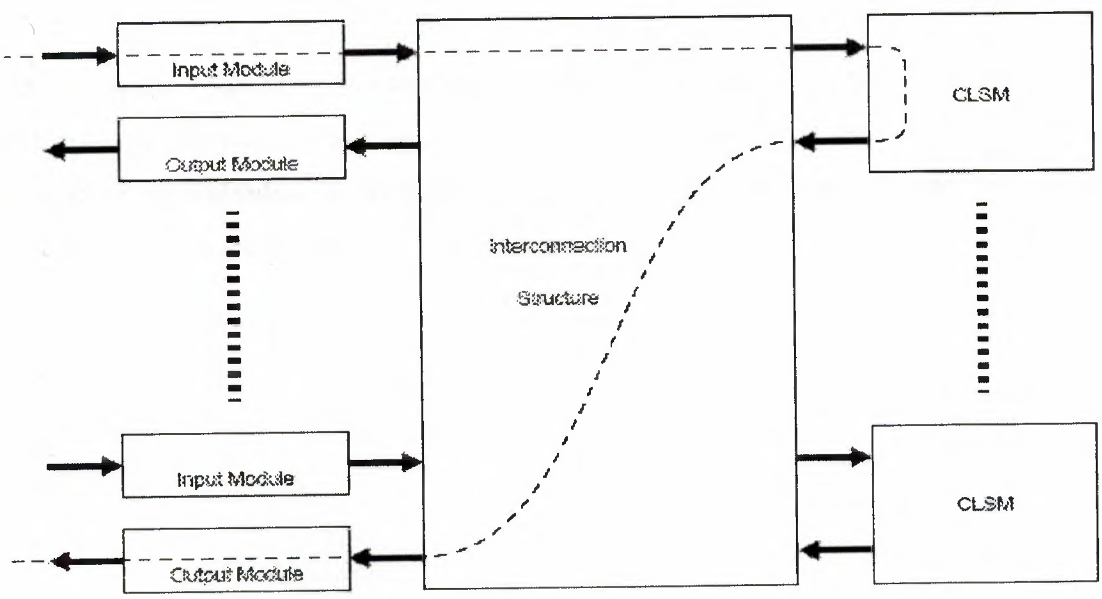


Figure 3-11: Implementation Architecture 3

outgoing ATM connection for the packet, using the address information contained in the packet. The packet is (virtually) segmented again, and the segments are forwarded to the Output Module, responsible for the outgoing link, via the IS. The Output Module constructs an ATM cell, which is transmitted on the outgoing link.

The architecture corresponding to this distribution of functionality does not enforce a one-to-one mapping between Input or Output Modules and Input respectively Output CLSMs (Figure 3-11). One Input and one Output Module is needed per ATM link. The number of CLSMs used in a CLS can depend on the expected traffic load.

This implementation architecture allows for a flexible dimensioning of the CLS. The Input and Output Modules should be dimensioned in such a way that they can cope with the maximum cell rate on a link. Furthermore, the total number of CLSMs should be such that the CLS can achieve the required packet throughput with the required QoS.

A disadvantage of this architecture is that all data is switched by the IS twice. Another disadvantage is that different CLSMs may have to send cells to the same outgoing ATM connection, because the cells belong to packets that are destined for the same CLS.

As a result, coordination between CLSMs is required for traffic shaping and multiplexing on that connection, or several connections to the same CLS should be maintained. We introduce another implementation architecture, which avoids the need for coordination or maintaining several connections.

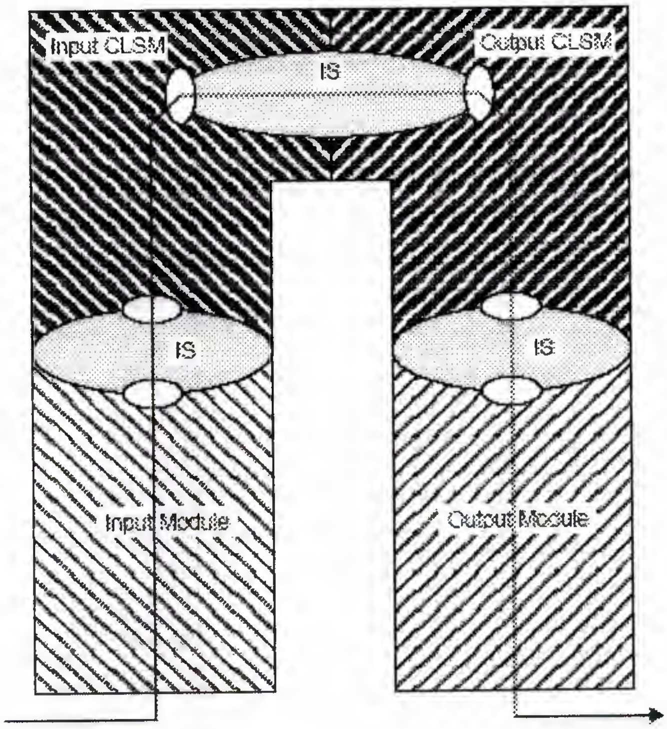


Figure 3-12: Distribution of Functionality (Architecture 4)

3.1.5 Architecture 4

According to Architecture 4, the functionality of the AAL and BCL Network Layer is implemented in separate modules for input and output, i.e., Input CLSMs and Output CLSMs. Input Modules, Output Modules, Input CLSMs, and Output CLSMs are all connected to a central IS (Figure 3-12). Each incoming ATM connection is handled by a single Input CLSM, and analogously each outgoing ATM connection is handled by a single Output CLSM.

ATM cells are processed by the Input Module where they first arrive. Based on the incoming ATM connection, the payloads are forwarded, via the IS to the proper Input CLSM. After (virtual) reassembly and network layer processing, an outgoing link and outgoing ATM connection are determined by the routing function of the Input CLSM. The packet and an indication of the required outgoing connection are forwarded via the IS to the Output CLSM responsible for the outgoing ATM connection. This Output CLSM maintains state information for each outgoing connection it handles. It performs the necessary network layer processing, and (virtually) segments the packet, and forwards it to the Output Module that is connected to the proper outgoing link.

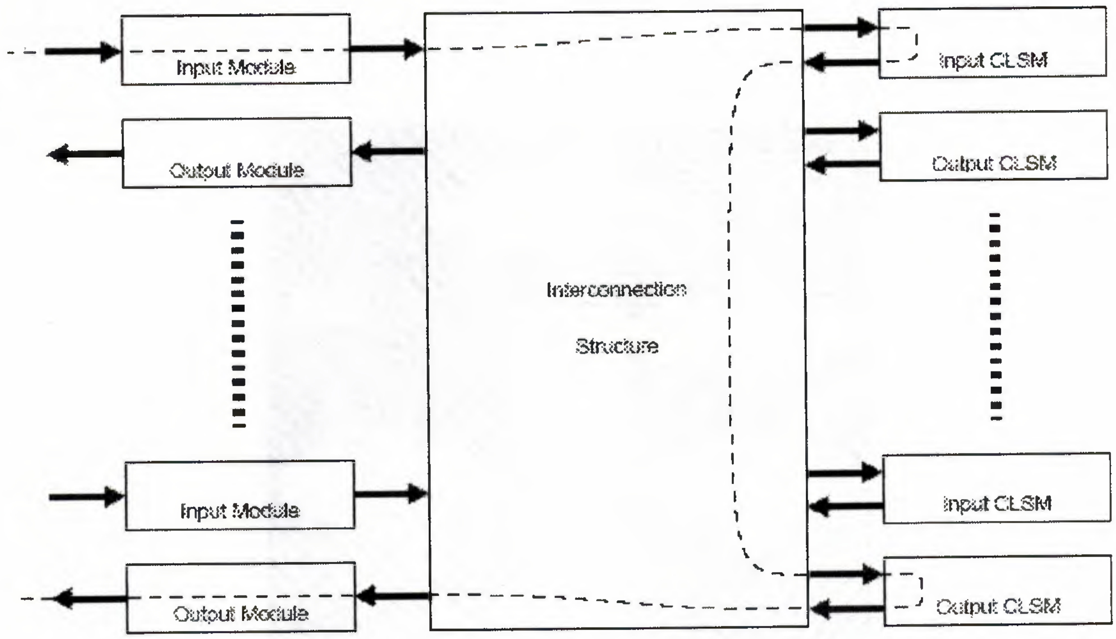


Figure 3-13: Implementation Architecture 4

This Output Module generates new ATM headers, and transmits the cells on the ATM link. This distribution of functionality allows, like Implementation Architecture 3, a flexible mapping of ATM links (and Input and Output Modules) to Input and Output CLSMs. Furthermore, it allows, like Architecture 2, for each CLS or endsystem that is connected to the system, to cooperate with a single Input CLSM and a single Output CLSM. Figure 3-13 shows the architecture that will result from the distribution. This implementation architecture seems to combine the advantages of architectures 2 and 3. However, it has one important disadvantage. All data is switched by the IS three times. In order to solve this problem, another implementation architecture is identified.

3.1.6 Architecture 5

According to Architecture 5, all AAL and BCL Network Layer functions for a packet are performed in a single CLSM, except for some specific functions, which are performed in a separate module called Output Packet Processing Module (PPM). These are functions that maintain state information, which is specific for the next node, a packet will be forwarded to. By performing these functions in the same Output PPM for all the packets that should be forwarded to that next node, it can be avoided that several copies of the

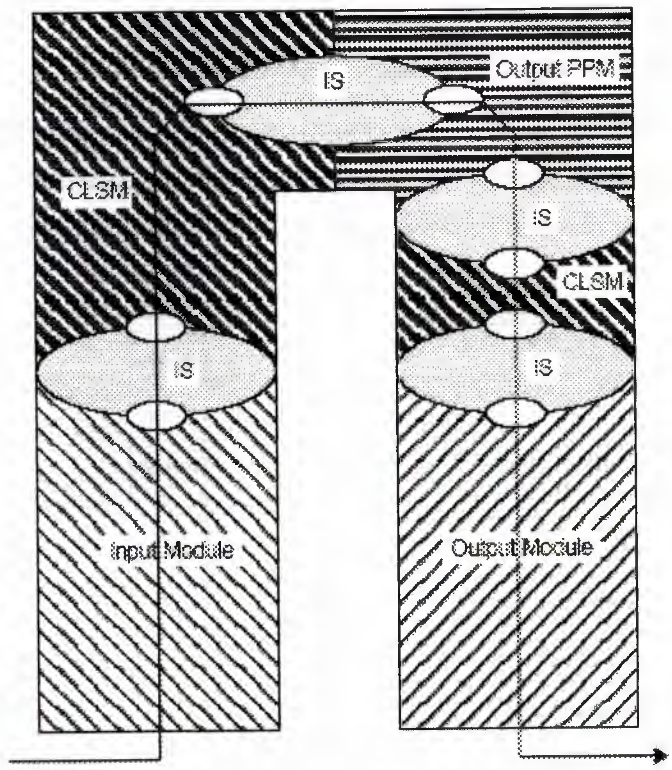


Figure 3-14: Distribution of Functionality (Architecture 5)

information has to be maintained in different modules. An example of such a function is source address screening. By performing this function always in the same Output PPM, it can be avoided that copies of the list of allowed source addresses for a certain destination end-system have to maintained in several modules. For its operation, an Output PPM only needs the information contained in the first cell of a packet, and the result can be returned to the CLSM in a single switch data unit. Input Modules, Output Modules, CLSMs, and Output PPMs are all connected to a central IS (Figure 3-14). ATM cells are processed by an Input Module upon arrival. Depending on the incoming ATM connection

(VPI/VCI) their payload is forwarded to a certain CLSM. This CLSM maintains state information for all connections it terminates.

Here, AAL processing takes place, and a packet is (virtually) reassembled. Network layer processing for incoming packets takes place, and routing is performed. After the routing decision has been taken, the network layer header, the CPCS header, and an indication of the required outgoing connection are forwarded to a specific Output PPM, depending on the required outgoing ATM connection. This Output PPM maintains state information for all ATM connections that originate there. In the module, processing for outgoing packets and possibly AAL multiplexing is performed. The result, e.g., multiplexing information (an MID value) is returned to the CLSM in a single switch data unit. Upon receipt of this

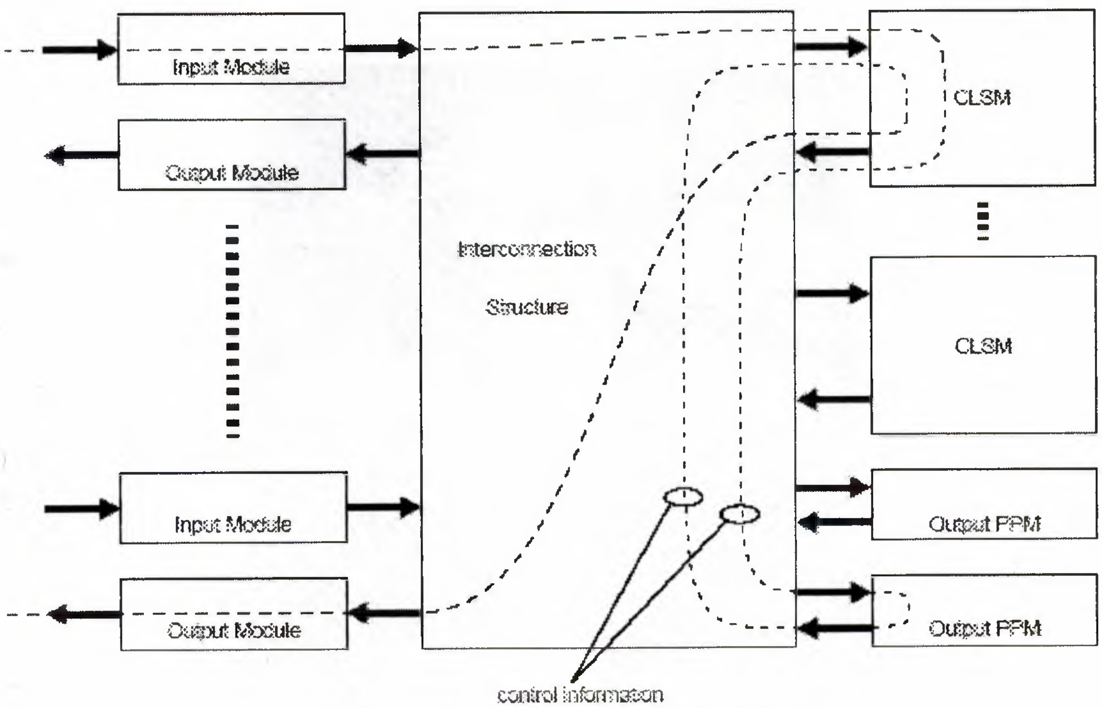


Figure 3-15: Implementation Architecture 5

result, the CLSM (virtually) segments the packet again, performs additional AAL layer processing, and forwards the cells via the IS to the appropriate Output Module. This module performs ATM layer processing, and sends the cells out on the outgoing link.

Figure 3-15 shows the implementation architecture for this distribution of functionality.

In this architecture, there is a flexible mapping between Input and Output Modules and CLSMs. Furthermore, all the output processing for a single end-system can be performed in the same Output PPM. However, since only header information is forwarded from the CLSM to this module, unnecessary switching of data can be avoided.

3.1.7 Architecture 6

Routing is a very demanding function that has to be performed in a CLS. It involves looking up the destination address in a possibly large routing table. It is a potential performance bottleneck of a CLS. Therefore, it may be wise to remove the one-to-one mapping of CLSM to routing function from the implementation architecture, in order to remove the bottleneck. This leads to the following implementation architecture.

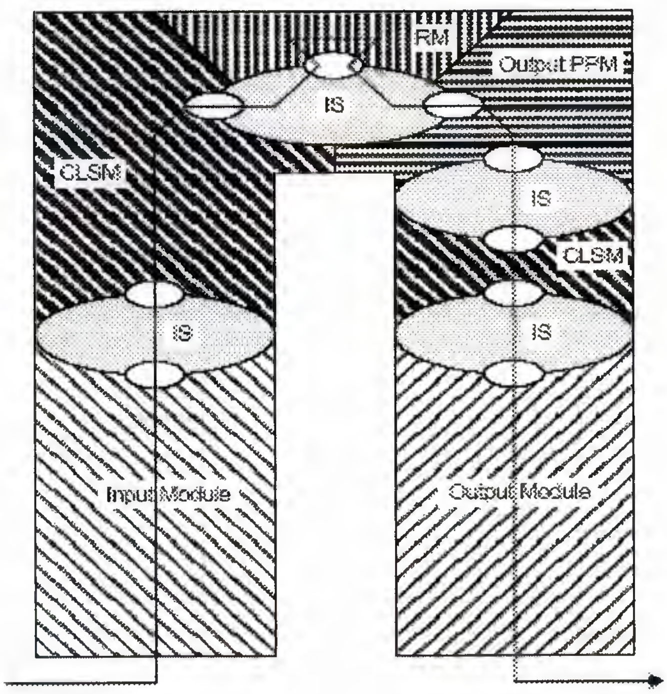


Figure 3-16: Distribution of Functionality (Architecture 6)

According to Architecture 6, routing is performed in separate Routing Modules (RMs), which are connected to the IS. Furthermore, Input and Output Modules, CLSMs, and Output PPMs are connected to the central IS (Figure 3-16).

Incoming ATM cells are processed by an Input Module, and their payload is forwarded to one of the Input CLSMs, depending on the incoming ATM connection. Here, a packet is (virtually) reassembled from the incoming cell stream. After (or during) performing network layer input processing, the network layer header and the CPCS header are forwarded to one of the RMs. The decision, which RM to forward the information to is taken randomly, or based on the load of the RMs. No state information related to ATM connections is maintained here.

This module determines the outgoing ATM connection and link, based on the network layer destination address. This information, together with the received information, is now forwarded to the Output PPM responsible for the outgoing ATM connection. After performing additional network layer processing for the output, a switch data unit with an identifier for the outgoing ATM connection and possibly multiplexing information is returned to the CLSM. The CLSM (virtually) segments the packet again, performs AAL processing, and forwards the cells of the packet to the Output Module responsible for the outgoing link. This module constructs the ATM cell header, and transmits the ATM cell on the outgoing link (Figure 3-17).

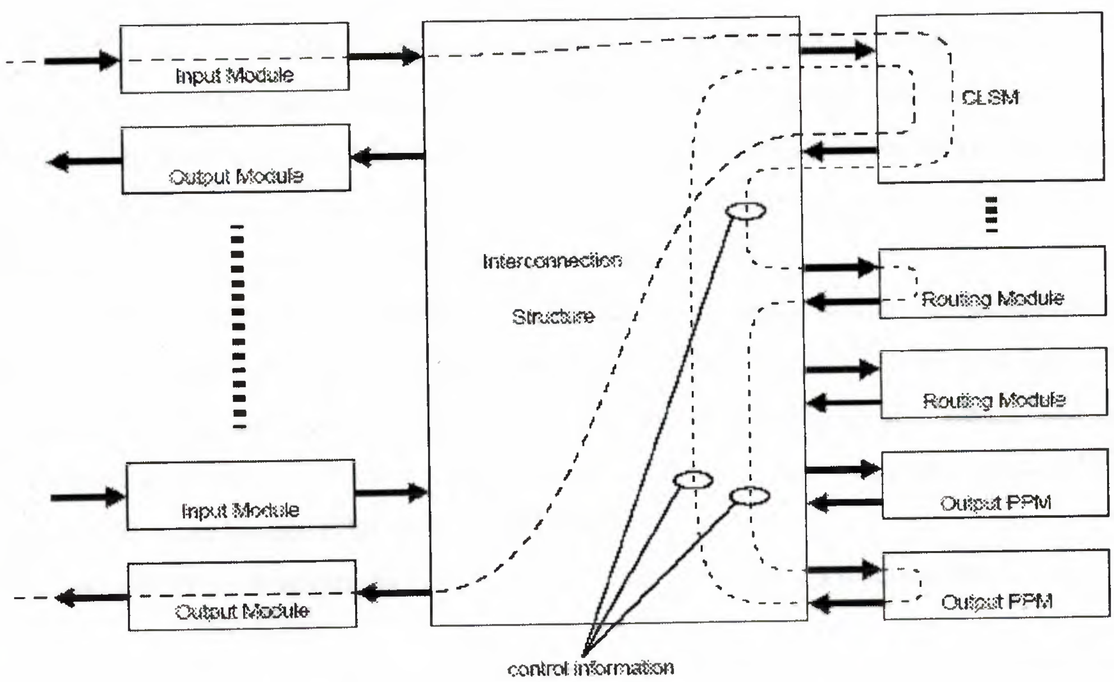


Figure 3-17: Implementation Architecture 6

Like the previous implementation architectures, this one allows a flexible mapping between links and CLSMs, only cells belonging to the same ATM connection on a link have to be forwarded to the same CLSM. Furthermore, it allows a flexible mapping between CLSMs and the routing function, performed in a separate RM. The decision, which RM to forward a packet to can be taken randomly per packet, or depending on the load of the RMs. This modularity of the implementation allows for an optimal balance between the processing capacities for the different functions, i.e., the number of modules performing the same function in parallel can be chosen such that the throughput for each group of parallel modules is as desired.

3.2 Functional Description of the Modules

In the previous subsection, we have identified a number of modules, which will implement the functions that need to be performed in a CLS. In order to provide more insight in the distribution of functionality in a CLS, we will now describe the functions to be performed in each of the modules in more detail.

Depending on the implementation architecture, not all modules are present in every design. If an RM or an Output PPM is not present in architecture, its functionality is assumed to be present in CLSMs. Furthermore, we describe the functionality of both an Input and an Output CLSM, while we often do not distinguish between the two. In that case, the CLSM is supposed to have the functionality of both.

First, the IS is described in Section 3.2.1. The Input and Output Modules are described in Section 3.2.2 and Section 3.2.3 respectively. The CLSM is described in Section 3.2.4. Its functionality can be split into an Input CLSM and an Output CLSM, which will be described in Section 3.2.5 and Section 3.2.6. Alternatively, part of its functions could be performed by an Output PPM and possibly a RM.

We end with their descriptions in Section 3.2.7 and Section 3.2.8 respectively.

3.2.1 Interconnection Structure (IS)

The sole purpose of the IS is to move information between modules. The main functions that can be identified in the IS are switching and buffering.

Interactions between the IS and other modules constitute of exchanging switch data units. A switch data unit consists of control information and payload. A data unit contains at least an indication of the required destination module when it enters the IS. When it leaves the IS, it contains at least an indication of the source module. The size of the payload is such that it can contain the parameters of an ATM-DataRequest or ATM-DataIndication primitive and an internal connection identifier. Alternatively, the IS can be implemented in such a way that a number of switch data units need to be exchanged for the transfer of the parameters of a single primitive.

3.2.2 Input Module

The Input Modules of a CLS have the same functions as the Input Modules of an ATM switch. They connect to the links to other nodes in the network. The functionality consists of Physical Layer functions and ATM functions. Furthermore, if the Input Module is connected to an IS, conversion of ATM cells to switch data units is performed.

Physical Layer functions include line termination, cell delineation, and Header Error Control (HEC) field verification. ATM Layer functions can include translation of VPI/VCI, routing tag insertion, rate adaptation, and possibly Usage Parameter Control (UPC).

A management module will provide an Input Module with a VPI/VCI mapping table. In this table, it can look up the VPI/VCI of an incoming ATM cell, and obtain an internal connection identifier and also the routing tag for the CLSM the cell has to be forwarded to.

3.2.3 Output Module

The Output Modules of a CLS have the same functionality as those of an ATM switch. Similarly to an Input Module, an Output Module has to perform Physical Layer functions, ATM Layer functions, and conversion of ATM cells from the IS format.

Physical Layer functions include line termination, HEC field generation, and transmission frame generation. ATM Layer functions can include VPI/VCI generation, rate adaptation, and buffering.

An Output Module maintains a table where it can look up for each internal connection identifier, which VPI/VCI value it has to give to the corresponding field in the ATM cell header. This table will be updated by a management module.

3.2.4 Wireless Service Module (CLSM)

The functionality of the CLSM, as present in implementation architectures 1 and 2, equals that of both the Input CLSM (Section 3.2.5) and the Output CLSM (Section 3.2.6) as described below. In implementation architectures 5 and 6, the functionality of the Output PPM (Section 3.2.7) and the functionality of both the Output PPM and the RM (Section 3.2.8) is excluded from the CLSM respectively.

3.2.5 Input CLSM

In Section 3.3.4, we have identified two different types of CLSs, Access CLSs, and Transit CLSs. The functions to be performed by an Input CLSM depend on the peer entity it communicates with. If the peer entity is another CLS the Input CLSM has to perform only transit functions. Otherwise, if the peer entity is a system outside of the public part of the network (in case of an Access Server), e.g., an end-system or an IWU, the Input CLSM will have to perform access functions as well as transit functions.

The following transit functions have to be performed by an Input CLSM:

• (Virtual) Packet Reassembly

The Input CLSM should be able to reconstruct the packet (CLNAP-PDU or CLNIP-PDU). This does not imply that the entire contents of the packet needs to be present in a CLSM at a certain moment in time. However, the Input CLSM, RM, and Output CLSM should be able to consult the PCI of the packet.

• Packet Integrity Verification

A limited amount of errors due to bit errors, cell los, missequenceing, or insertion can be detected, depending on the type of AAL used. The Input CLSM should detect these errors, and take the appropriate measures (e.g., deletion of entire packets if one of its segments is corrupted or lost).

• The functions listed for the RM (Section 3.2.8)

If needed, the following access functions will be performed by the Input CLSM:

• Address Validation

The Input CLSM should check the source address specified in the packet against a table of source addresses that is allowed for the particular user. The user can uniquely be identified by looking at the incoming link, and the incoming VPI/VCI.

• Destination Address Screening

The Input CLSM should compare the destination address of the packet with a table of allowed addresses for the given source address.

• Access Class Enforcement

The rate at which data is arriving from a user should be checked against the rate corresponding to the agreed access class. A number of tables are to be maintained in an Input CLSM. A table with allowed source address values for each internal connection identifier is used by the address validation function. A table of allowed destination address values for each source address is used by the destination address screening function.

Furthermore, a routing table, containing for each destination address, an identifier for the Output Module and an internal connection identifier is present in the Input CLSM. The tables are downloaded and updated by a management module.

3.2.6 **Output CLSM**

For the Output CLSM, we can also distinguish between transit functions and access functions. The following transit functions need to be performed:

• **Cell Payload Construction**

The Output CLSM should transfer the received packets in segments that fit in ATM cells. In general, this will be the same segments that are received by the Input CLSM, but depending on the AAL type used, some of the AAL PCI has to be modified. If packets are really reassembled in the Input CLSM, also packet segmentation has to be performed.

• **Traffic Shaping**

Associated with an outgoing ATM connection, is a traffic descriptor, as part of a traffic contract, which characterizes the traffic that can be expected on the connection. This descriptor is used by the ATM network to reserve the proper resources for a connection. The parameters of this traffic descriptor are also used by the traffic shaping function, to shape the outgoing cell stream on a connection, so that the traffic contract is not violated. Cells that would violate the contract are delayed until they can be safely transmitted on the connection.

• The transit functions listed for the Output PPM (Section 3.2.7). The access function to be performed in an Output CLSM is the one listed for the Output PPM. The status information to be maintained in the Output CLSM includes that listed for the Output PPM. Furthermore, parameters for the traffic shaping function need to be maintained.

3.2.7 **Output Packet Processing Module (PPM)**

A transit function that may be performed in an Output PPM is the following:

• **Multiplexing (Packet Interleaving).**

In order to allow the interleaving of several packets on an outgoing connection, the Output CLSM can multiplex segments from different packets. For this purpose, the MID field has been defined in the AAL 3/4 protocol. AAL 5 does not allow for the interleaving of several packets on an outgoing VC connection. However, it allows for interleaving on an outgoing VP connection, where the VCI is used for packet identification.

Furthermore, the following access function should be performed:

• Source Address Screening

The Output CLSM should compare the source address of the packet with a table of allowed addresses for the given destination address. The state information to be maintained in the Output PPM includes a table of used MIDs (VCIs in case AAL 5 is used over VP connections), which is used by the multiplexing function. This table is updated by the PPM. For the source address screening function, a table of allowed source addresses for each destination address is maintained in the Output PPM. The table is updated by a management module.

3.2.8 Routing Module (RM)

The RM has only a single function:

• Selection of outgoing ATM connection

Based on the destination address given in a packet, the RM will determine the ATM connection (VP/VC) and link on which the packet has to be transferred. For this function, a routing table will be maintained, indicating for each destination address which Output Module, and which internal connection identifier should be used. The table will be downloaded and updated by a management module.

3.3 Implementation Issues

Given implementation architecture for a CLS, a number of important issues need to be addressed before detailed implementation decisions can be made. They will be addressed below. First, in Section 3.3.1, the reassembly of packets in the CLS will be discussed. In Section 3.3.2, it will be discussed how copying of data in the CLS can be avoided. In Section 3.3.3 we will discuss how the buffering of SAR-PDUs can take place. Finally, in Section 3.3.4, we discuss the issue of integrating a CLS in an ATM switch.

3.3.1 Packet Reassembly

An important issue in the design of the CLS is the question whether or not a network layer packet should be completely reassembled in the server. In principle, the BCL Network Layer, which is implemented in the CLS, operates on packets. For that reason, this packet should be reassembled from the SAR-PDUs, which are transferred in ATM cells. Before transmitting the packet on the outgoing link, it needs to be segmented again, in order to be able to fit into the SAR-PDUs, which are to be transferred in ATM cells. This mode of operation, where packets are completely reassembled before transmission,

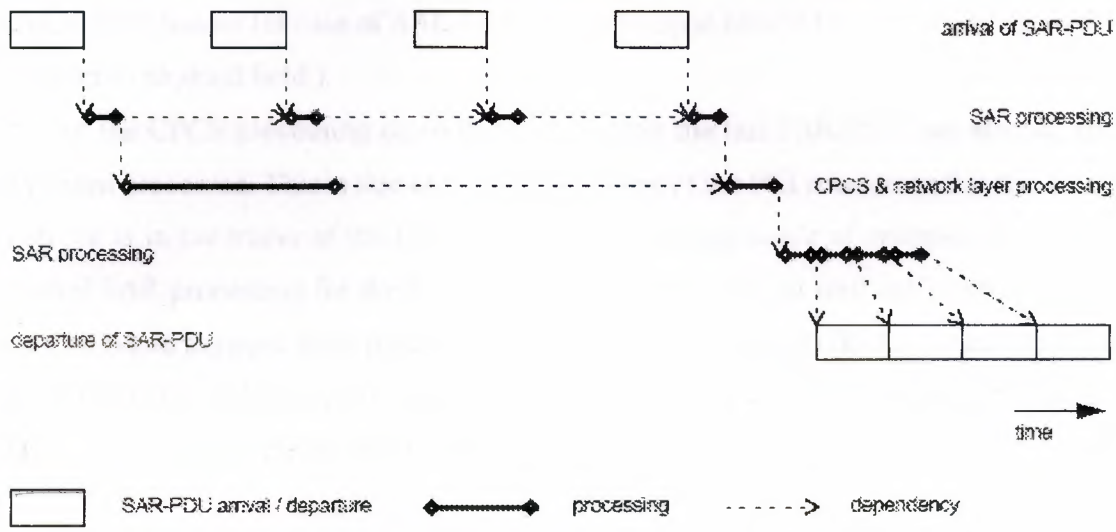


Figure 3-18: Dependencies between Processing in Case of Message Mode

is called *message mode*.

As an alternative to complete reassembly, the packet may be processed on a per cell basis. Upon receipt, cells are processed one by one, and transmitted again. State information is maintained in the CLS, in order to be able to process subsequent cells of the same packet. This mode of operation is called *streaming mode*.

The two modes correspond to the modes of refining the AAL service primitives, discussed in Section 2.4.2.

Reassembly of packets in each intermediate system (e.g., CLS) increases delays, and is inefficient in terms of required buffer capacity. However, there can be a requirement for the CLS to operate in message mode. For specific functions, the processing needs to be completed before the first cell of the packet can be sent out. Such a function may need the presence of the complete packet. If this is the case, message mode is the only suitable mode of operation. Figure 3-18 shows the dependencies (by means of dashed arrows) between the arrival of SAR-PDUs, completion of the AAL and network layer processing, and the departure of the SAR-PDUs in the case where message mode is required. In general, the CPCS and network layer processing can start as soon as the first cell of a packet has been received, since the CPCS-PDU header and the network layer PCI is contained in this cell (ITU-T Recommendation I.364 specifies a PDU header of 20 octets with a possible 20 octets extension for the CLNAP ([81]). Together with the 4 octet CPCS-PDU header (in case of AAL 3/4) this makes up at most 44 octets, which fits in the SAR-PDU payload field.).

Part of the CPCS processing can only be done after the last SAR-PDU has arrived, and has been processed. This is due to the fact that some of the PCI is contained in this SAR-PDU (it is in the trailer of the CPCS-PDU). The message mode of operation should be used if SAR processing for the first outgoing SAR-PDU has to wait until completion of all CPCS and network layer processing, as is shown in Figure 3-18. As a result, the first SAR-PDU of a packet can only depart after the last one has arrived. Two possible reasons for this dependency can be identified:

1. Error handling for the incoming packet (i.e., error detection and discarding of erroneous packets) may require the presence of all SAR-PDUs of the packet. In order to avoid partial transmission of erroneous packets, error detection for the entire packet should be completed before header construction and transmission of the first SAR-PDU of a packet can take place. Otherwise, the first couple of SAR-PDUs of a packet could have been sent already at the moment the error is detected. Furthermore, if AAL 5 is used, errors in the address field are only detected after the AAL header analysis for the last SAR-PDU of the packet, since the CRC code for the entire packet is contained in the CPCS-PDU trailer in the last SAR-PDU.

2. In some cases SAR-PDUs of a packet cannot be transmitted interleaved on the outgoing connection with SAR-PDUs belonging to other packets. In this case, a CLS should wait with transmitting a cell until the complete packet it belongs to has been

received. Otherwise, delay or loss of one of the subsequent SARPDUs of that packet would cause excessive delays to other packets, since their transmission has to wait until the last SAR-PDU of the previous packet has been transmitted. Cells of different packets cannot be interleaved on an outgoing connection if AAL 5 is used over VC connections because AAL 5 does not have a multiplexing (MID) field.

The streaming mode of operation is likely to result in an improved performance, compared to the message mode. Operation in streaming mode is only possible, if the dependency described above does not exist. In this case, AAL header construction and transmission for the first cell of a packet can start as soon as the network layer processing related to the CPCS-PDU header and the network layer has been performed. Figure 3-19 shows the dependencies between the arrivals of AAL-PDUs, completion of the AAL and network layer processing, and the departure of the AAL-PDUs in this case. From the figure, it can be seen that the SAR processing for the first outgoing SAR-PDU can start as soon as the CPCS and network layer processing performed upon its arrival has finished. It does not have to wait until the CPCS and network layer processing related

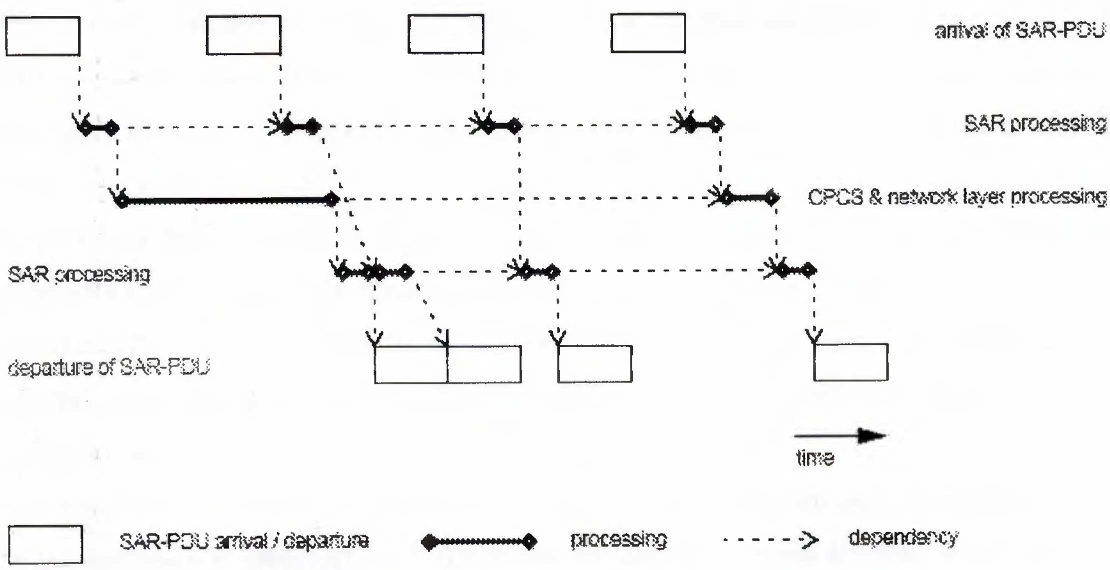


Figure 3-19: Dependencies between Processing in Case of Streaming Mode

to the last SAR-PDU of a packet has completed.

In streaming mode, an extra SAR-PDU routing table, containing information about the route of all packets that are being transferred, needs to be maintained.

For the first SAR-PDUs of the packet, the outgoing ATM connection, and in case of AAL 3/4, the outgoing MID value is determined, using the main (packet) routing table. This information is stored in the SAR-PDU routing table, in order to be able to use it for consecutive SAR-PDUs of the packet. For SAR-PDUs, which are not the first one of a packet, the outgoing ATM connection and possibly MID is looked up in the SAR-PDU routing table. Furthermore, in case a SAR-PDU is the last one of the packet, the entry for that packet is removed.

Disadvantages of the streaming mode are that it can only function well if interleaving is possible, and that partially corrupted packets are proceeding through the network. It is still an open issue whether the expected performance advantage of the streaming mode is outweighed by the disadvantages.

3.3.2 Data Copying

Most of the functions that need to be performed in a CLS do not need the presence of the user information in the packet. Instead they only need to know some of the PCI of the concerned protocol layer. If one of the modules is able to extract the appropriate information from the incoming cell stream, send it to other modules, which process the PCI, and buffer the cell while waiting for the result of the operation, it can be avoided that data is copied too often. Furthermore, in this way some of the functions can be performed simultaneously.

Copying of data is undesirable in a CLS. Transferring all cells of a packet from one module to another module involves an additional delay, especially when the cells are transferred via the IS. Furthermore, if all cells of a packet are transferred using the IS several times, the required throughput of the IS increases linearly with the number of times the data is transferred.

The need for data copying depends on the distribution of functionality over the modules in implementation architecture. If functions that need to process all cells of a packet are concentrated in a few modules, copying can be avoided to a large extent. These modules can extract the information that is needed by other modules (e.g., address information) and transfer it to them. The latter modules in turn need to return only the result of their processing.

Of course Input and Output Modules operate on all the cells, since these are only concerned with cells, not with packets. The cells enter an Input Module on a link from another CLS, or from an end-system. They have to depart from an Output Module on a link to another CLS, or to an end-system. Somehow, these cells need to be transferred from an Input Module to an Output Module. Since additional processing is needed, i.e., AAL and network layer processing, at least one other module should be involved. Architecture 1 employs this basic scenario. However, because of reasons given in Section 3.1, an IS is often needed to interconnect the modules. Furthermore, it turned out to be desirable to connect Input and Output Modules directly to the IS.

Taking the above into account, we still have a number of additional constraints to deal with in determining the distribution of functionality that is most optimal with respect to the amount of data copying. The first one is related to the information contained in the packet (either PCI or payload) and packet related results from other functions that are needed by a certain function. Clearly, both the needed information from the packet and the needed results from other functions should be available in the module where the function is implemented. Table 3-1 shows for each of the AAL and network layer functions to be implemented in the CLS, what information it needs, and what information it results in. Let us consider the source address screening function as an example to illustrate the table. The only information these functions needs of a packet are the source and destination address. This information should be obtained from the (virtual) packet reassembly function. The result of the source address screening is an indication whether or not the packet should be forwarded (ok / not ok). This information will be used by the cell payload construction function, which should only process a packet if the result of the address screening is "ok".

Table 3-1: Needed and Resulting Packet Related Information for Functions

function	needed information	resulting information
(virtual) packet reassembly & packet integrity verification	all AAL-PDUs + incoming ATM connection	packet (incl. address information)
address validation	source address + incoming ATM connection	ok / not ok
destination address screening	source address + destination address	ok / not ok
access class enforcement	source address + packet length	ok / not ok
selection of outgoing ATM connection	destination address	outgoing ATM connection
source address screening	source address + destination address	ok / not ok
multiplexing	outgoing ATM connection	multiplexing information
cell payload construction	packet + ok / not ok multiplexing information	AAL-PDUs
traffic shaping	number of AAL-PDUs	departure times for AAL-PDUs

Avoiding unneeded copying of data implies that it should be avoided that the complete packet is needed in several modules. Therefore, it would be wise to implement the (virtual) packet reassembly and packet integrity verification and the cell payload construction in a single module. This is exactly what has been proposed in implementation architectures 1, 3, 5, and 6.

The second additional constraint comes from the fact that certain information has to be shared for the processing of different packets. This information can be static, e.g., the allowed source addresses for a destination address, which are used for source address screening. It can also be dynamic, i.e., it is updated during the processing of a certain packet. This is for instance the case with traffic shaping, where traffic statistics have to be maintained for an outgoing connection, which have to be updated for each packet. Table 3-2 shows for each of the AAL and network layer functions, which shared information it needs for its processing, which shared information it results in, and which packets' processing operates on the same shared information. Let us consider the source address screening function again. The function needs to have a table of allowed source addresses,

Table 3-2: Needed and Resulting Shared Information for Functions

function	needed shared information	resulting shared information	shared by all packets with the same
(virtual) packet reassembly & packet integrity verification	multiplexing and reassembly info	multiplexing and reassembly info	incoming ATM connection
address validation	allowed source addresses	-	incoming ATM connection
destination address screening	allowed destination addresses	-	source address
access class enforcement	access class + traffic characteristics	traffic statistics	source address
selection of outgoing ATM connection	outgoing ATM connection	-	destination address
source address screening	allowed source addresses	-	destination address
multiplexing	multiplexing info	multiplexing info	outgoing ATM connection
cell payload construction	-	-	
traffic shaping	traffic statistics	traffic statistics	outgoing ATM connection

which is used for the processing of all packets with the same destination. In order to avoid maintaining several copies of shared information, it would be best to implement all functions using the same shared information in a single module. As an example, it would be best to implement source address screening for a specific destination in a single module. This is not always possible, mainly because of scalability reasons. A more severe constraint comes from the sharing of dynamic information between packets. Implementing functions operating on the same information in different modules would require complex synchronization between the modules in order to let each module have the most up-to-date information.

This would for instance be necessary if traffic shaping for the same outgoing ATM connection would be performed in several modules. Since this synchronization will be very hard to realize, an outgoing ATM connection should be used by only a single (Output) CLSM. This implies all traffic destined for a certain CLS or end-system should depart from the same CLSM, or that several ATM connections to this CLS or end-system should be maintained. The former option is only possible in Architectures 2 and 4 (and in Architecture 1, which has only a single CLSM). The other implementation architectures should maintain several outgoing connections to the same CLS or end-system.

It can be concluded that there is often a trade-off between avoiding data copying, and designing a flexible large-scale implementation. It is still an open issue which architecture has the best overall suitability. In fact, the choice for specific implementation architecture depends heavily on the situation where it is applied.

3.3.3 Buffering

Related to the problem of data copying is the problem of data buffering. At least one copy of the network layer packet (or a segment of it, depending on the mode of operation) needs to be buffered until all the needed processing has been performed. While operations are being carried out on part of the data (e.g., the destination address), data needs to be buffered until the results of the operation are available. It is most logical to buffer the data in the module where (virtual) reassembly of the network layer packet is performed, since all the data is available there anyway. The data also needs to be present in the module where the cell payload construction is performed, but this module is only known after part of the processing (routing) has already been performed (if it is not the same module).

If the CLS is operating in message mode, buffering of data for reassembly needs to be performed anyway. The reassembly buffer can also be used to store data until the results from the network layer processing are available. Also, the buffering for the traffic shaping function could be done in this buffer.

For a buffer operating in streaming mode, no reassembly buffer is needed. However, a buffer needs to be installed to store cells until processing has been completed. Different buffer strategies can be applied in this case. Data can be stored in a random access memory until the processing for the packet has been completed, and then be retrieved. Alternatively, cells can be stored in a FIFO (First In First Out) buffer for a fixed amount of time. After this time, the cell is retrieved from the buffer. Normally, the result of the packet processing is available at this time, and the cell is forwarded. In the exceptional case that the packet processing has not yet been completed, the cell, and all other cells from the packet are considered lost. If FIFO buffering is applied, an additional buffer is needed to store cells until the traffic shaping function schedules them for departure. The buffering strategy to be applied depends on the used mode of operation. In case of streaming mode, several strategies are possible. An important advantage of FIFO buffering is that it does not require complicated buffer management schemes. However, it can on the other hand lead to additional delays and losses.

3.3.4 Integration with ATM Switch

Another implementation issue of interest is if, and to what extent, a CLS can be integrated with an ATM switch. Some of the functions to be performed in a CLS should also be performed in an ATM switch. These functions are those of the physical and ATM layer and switching and buffering. A CLS is integrated in an ATM switch if some of its functions are performed by modules that form also part of the ATM switch, i.e., that also handle regular ATM traffic.

What are the advantages of integrated implementation? Disregarding realization advantages (e.g., sharing power supplies), we can identify two types of advantages:

• Performance

The aim of providing a wireless service on top of an ATM network is that the ATM network functions as a cell transport network, and that the CLSs perform routing and switching of the packets transported in the cells. Therefore, a CLS will be connected to a single ATM switch, or, to guarantee a higher availability, to a small number of them. This implies that traffic leaves the CLS through the same ATM switch as through which it arrived. Integrated implementation avoids that ATM cells are processed and switched more often than is needed. This results in improved performance in terms of throughput and delay.

• Smooth evolution

Wireless services can be assumed to form one of the major applications in early ATM networks. As time proceeds, the demand for wireless services may be gradually overtaken by a demand for connection-oriented services.

Integrated ATM switches / CLSs can evolve by replacing only those modules that are specific to wireless traffic, i.e., the modules performing AAL and network layer functions.

In defining the modules of a CLS in Section 3.1, we have already taken into account that some of the functions are performed in an ATM switch as well, and that they can hence be implemented in identical modules. These modules where the Input and Output Modules, responsible for the physical and ATM layer processing, and the IS, responsible for switching and buffering of data. If we want to share the same modules by regular ATM traffic and wireless traffic, one of the modules has to be able to identify wireless traffic, and forward it to a module dedicated to this traffic. The IS is the most natural module for this, since its function is to switch, i.e., forward depending on some information, cells.

Therefore, the use of the same IS is essential for an integrated ATM switch / CLS.

In an ATM switch, this IS is called an ATM Switching Fabric. Besides the IS, also the Input and Output Modules can be shared between connection-oriented and wireless traffic. As far as the physical and ATM layer processing is concerned, there is no difference between the processing of both types of traffic. This more comprehensive integration is only possible if the Input and Output Modules are directly connected to the IS in the implementation architecture.

From the implementation architectures we have defined, all but the first one (which has no IS) allow for sharing the IS with an ATM switch. Architectures 3, 4, 5, and 6 can also share their Input and Output Modules with an ATM switch. Figure 3-20 shows the implementation architecture of a CLS (according to Architecture 6) with an ATM switch. Whether or not a CLS will indeed be integrated with an ATM switch depends mainly on considerations beyond the scope of this Project. If the wireless service is for instance operated by another organization than the underlying ATM network, there is a need for a clear interface between a CLS and an ATM switch. There will be a need for both integrated and separate CLSs.

3.4 Design Criteria applied to Implementation

In Chapter 1, we have identified a number of design criteria. The extent to which some of these criteria are met differs for different implementation architectures.

In this section different architectures are compared with respect to a number of criteria. The important design criterion of performance is discussed in separate chapters (Chapters 5 and 6). Here we discuss availability in Section 3.4.1 and scalability in Section 3.4.2.

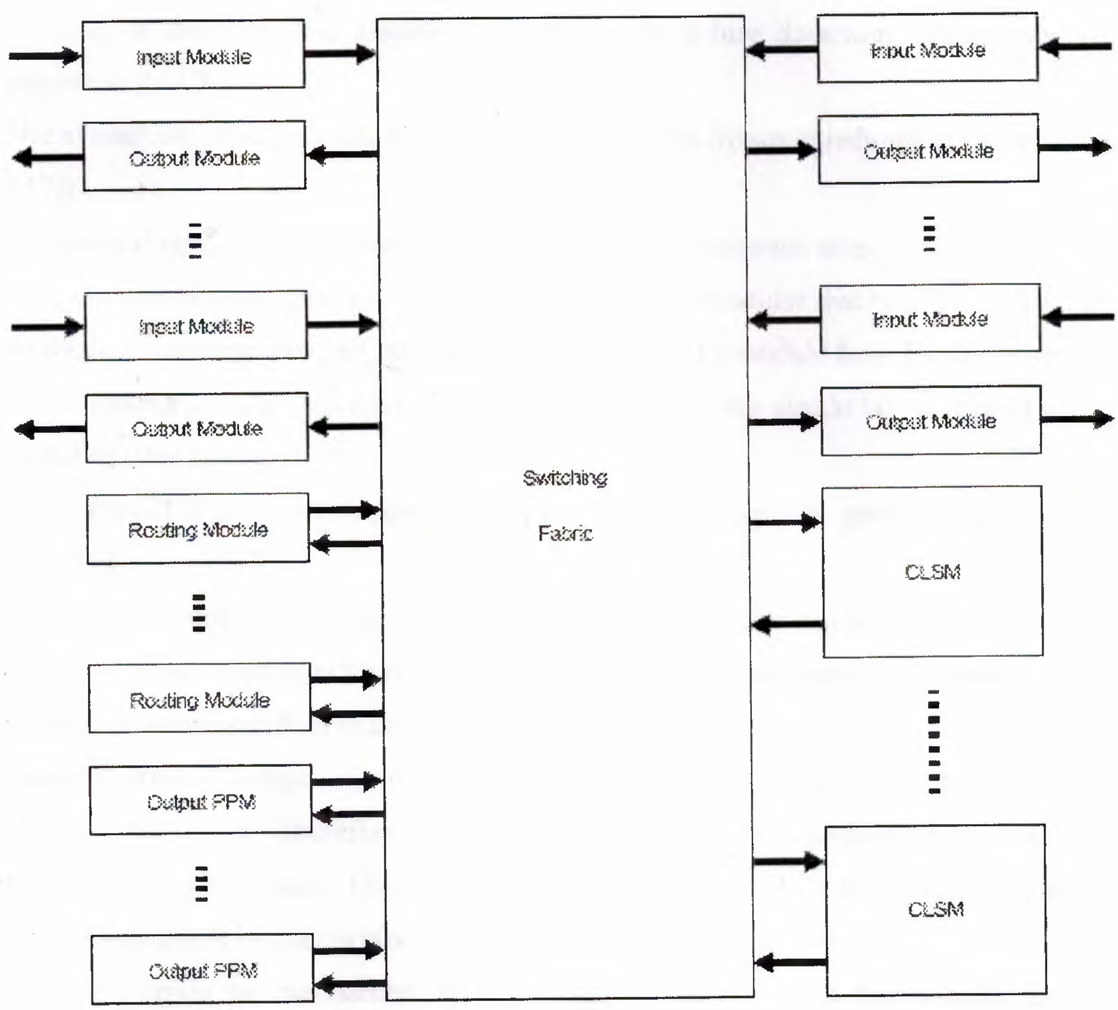


Figure 3-20: CLS integrated with an ATM Switch

3.4.1 Availability

Significant differences can be found in the ability of the different implementation architectures to cope with failing components. This ability is an important design criterion in the choice for a specific architecture. Here, the intent is to discuss qualitatively the inherent pros and cons of the architectures as far as their ability to cope with failure of the constituting modules is concerned. We do not have the intention to cover availability issues completely. Furthermore, we do not discuss the availability of the individual modules themselves.

We assume that modules either perform adequately or fail completely. A module exhibiting erroneous behavior can be shut down by management if the erroneous behavior is detected. We assume that appropriate failure detection mechanisms are present in the CLS.

The availability of a CLS can be enhanced by applying dynamic redundancy techniques ([47]).

The general idea of these techniques is to reconfigure the system after failure of one of its components has been detected. For a CLS consisting of modules that may fail, this would imply that other modules can take over the processing if a module fails. Furthermore, the status information that was maintained in the failing module should be transferred to the module(s) that take over.

Thus, the CLS as a whole can stay operational, although its performance may be degraded.

The ideal situation is where all modules are configured around a central IS, so that traffic can easily be rerouted to other modules by the IS, in case of module failure. Note that this scenario assumes a sufficient availability of the IS.

A module failing completely will not be able to transfer its status information to other modules any more. Therefore, one or more copies of this information should be maintained in other places. Three alternatives can be identified for maintaining copies:

1. One copy could be maintained centrally, by the management entities,

2. Copies could be maintained by all modules with the same functionality as the concerned module, or

3. A copy could be maintained by a dedicated module, which is designated to take over if the concerned module fails.

Using the first alternative implies that the status information is downloaded to the module that takes over in case of failure. This reveals the disadvantage of the alternative; an additional delay is introduced between failure detection and recovery. The second alternative does not suffer from this disadvantage, but has to maintain a lot of status information in a lot of modules. This disadvantage is removed in the third alternative, although this alternative is somewhat more vulnerable to simultaneous failure of several modules.

Maintaining copies of all status information would require much interaction between the concerned module and the module where a copy is maintained.

Since it is very likely that a number of cells get lost in case of module failure, it does not make sense to avoid loss of the other cells of the packets currently being processed. Therefore, only copies of static information, i.e., information that is not updated after each packet (see Table 3-2), should be maintained. Upon start of service of the redundant module, the dynamic status information can be reset to initial values.

In case of ATM link failure, the ATM network is assumed to perform a reconfiguration. This could imply that incoming and outgoing ATM connections access the CLS through different Input and Output Modules. In this case, it is advantageous, if these connections can still be handled by the same CLSM. For that reason it is advantageous not to have a fixed CLSM per Input or Output Module.

Considering the implementation architectures identified in Section 3.1, we can see that Architecture 1 does not allow for any reconfiguration, since it does not have any redundant modules. The other architectures do allow for reconfiguration, if redundant modules are installed. Architecture 2 is not very flexible in its reconfiguration, since each Input or Output Module is connected to a single CLSM, and not via the IS to all. Therefore, the architectures 3 through 6 can be considered most suitable for increasing availability if individual modules are not sufficiently reliable.

3.4.2 Scalability

It is important to know to what extent a CLS can be dimensioned to cope with the expected traffic demands. Since these demands will change in time, it must be possible to extend an already installed CLS. Note that we concentrate on the scalability of a single CLS here. It should, of course, also be possible to extend the overall overlay network by installing new CLSs.

The throughput of a CLS can be increased by installing additional modules for processing. However, since only a single IS is present in (most of) the implementation architectures, the IS could form a bottleneck for the scalability of the CLS. It is important to use the IS efficiently if a CLS is to be extended. On the other hand, some ISs will to a certain extent be scalable themselves. It can be expected that the number of modules that can be attached to an IS can be increased. This possibility depends on the type of IS that is used. All implementation architectures but the first one allow for extension by installing extra modules, as long as the IS capacity is sufficient.

Another aspect of scalability is the balance between the loads on different types of modules. Different types of modules perform different functions. The load on these modules depends on various parameters. For an Input or Output Module, the load depends on the number of ATM cells that should be processed per second. For a packet processing module (the Output PPM), the load depends on the number of packets that should be processed per second. The same applies to the RM, of which the load also depends on the size of the network, because of the size of the routing tables.

In a CLS, the ratio between the number of cells offered per second and the number of packets offered per second is unpredictable, and will vary during operation. It depends on the fraction of the overall traffic in a switch that is wireless traffic, and on the number of cells per packet. Furthermore, the network size will probably not grow proportionally with the traffic load. Therefore, it is very important for a CLS that parts that perform different types of functions can be scaled independently. In the above perspective, Architecture 6 is most flexible in terms of scalability.

Architectures 3, 4, and 5, which do not have a separate module for routing, are less flexible. Furthermore, Architecture 4 has the disadvantage that all cells are switched by the IS three times, which puts a heavy claim on the IS throughput.

Architecture 2 only switches all cells once, but has the disadvantage that the number of CLSMs (including routing functions) can not be dimensioned independently from the number of Input and Output Modules. Architecture 1 is not scalable at all.

3.5 Summary and Concluding Remarks

In this chapter, we have discussed the problem of implementing a CLS. We have come up with a number of different implementation architectures. These range from a very basic one, consisting only of a single Input Module, a single Output Module, and a single CLSM, to a large scale one, aiming at high performance, and consisting of a number of Input and Output Modules, CLSMs, RMs, and Output PPMs, interconnected by a central IS.

A number of specific issues related to implementation have been discussed. We can conclude that it is still an open issue whether the message mode or the streaming mode of operation is preferable.

Other implementation issues, i.e., data copying and integration with an ATM switch, are strongly related to the used implementation architecture. Table 3-3 summarizes these relations on a scale from 'bad', via 'poor' and 'reasonable', to 'good'.

Table 3-3: Summary of Comparison of Implementation Architectures

	avoiding data copying	integration with ATM switch	availability	scalability
Architecture 1	good	not possible	bad	bad
Architecture 2	reasonable	poor	poor	poor
Architecture 3	reasonable	good	good	reasonable
Architecture 4	poor	good	good	reasonable
Architecture 5	reasonable	good	good	reasonable
Architecture 6	reasonable	good	good	good

The table also summarizes the suitability of the architectures according to availability and scalability criteria. Performance aspects of the architectures are not included in the table. Those will be discussed in the next chapters.

CONCLUSION

In this Project, the design of a system for broadband **wireless** communications using the **Asynchronous Transfer Mode** has been addressed. The objective was to provide insight in some important aspects of the design, and to analyses design alternatives with respect to effectiveness, availability, scalability, and in particular, with respect to performance.

The Conclusion reviews the major achievements of this Project, and presents directions for further work. For a more detailed discussion of the results of the individual chapters, we refer to their concluding sections. In Section 4.1, we discuss the main conclusions and results. In Section 4.2, we indicate areas for further research.

A number of applications that have to be supported by the B-ISDN have a wireless nature. Furthermore, nowadays LANs, which ask for interconnection, use a wireless protocol at the network layer (IP). Therefore, there is a clear demand for the provision of a wireless service by the B-ISDN.

Because ATM, the underlying switching and multiplexing technique of the B-ISDN, provides a connection-oriented service, a mismatch exists between the required wireless service and the provided service. Therefore, additional protocols need to be used on top of ATM to provide also the required wireless service.

In Chapter 2, an architectural framework has been presented, which places the protocols to be used in perspective. Two possible network architectures have been identified. The first one employs the so-called indirect method, where end systems are connected by means of end-to-end ATM connections. The second one employs the so-called direct method, where end-systems are connected to CLSs in the B-ISDN, which route packets through a wireless overlay network to the proper destination.
The implementation of a CLS is one of the most difficult tasks in the design of the wireless service. It has to support very high throughputs, while performing relatively complicated functions on the packets to be routed.

In Chapter 3, candidate implementation architectures for a CLS have been identified. It has been shown that using some of these architectures, it is feasible to implement a highly available and scalable CLS integrated within an ATM switch.

The Directions for Further Research that Connection management have been identified as an important function in a CLS or end-system. The OCDR mechanism has been proposed for the control of end to end connections, in case of the indirect method, and for the control of the connections between CLSs and end-systems. Other mechanisms for the control of connections between CLSs, on which traffic from numerous application processes will be multiplexed, should be further investigated. The VBC mechanism, which has been shown to be promising, needs to be analyzed thoroughly.

In our analyses, we have considered the performance of a single node. An interesting direction for further research is the performance analysis of a network of nodes. Moreover, it is interesting to develop a tool that can be used to dimension an entire wireless overlay network on top of ATM. Such a tool should support decisions regarding the switches in the ATM network, to which a CLS must be attached, the connections to be maintained between CLSs, and the bandwidth to be reserved on these connections.

The current ITU specifications of ATM have been used as a starting-point for our work. ATM, as it is defined now, is not the most suitable underlying communication system one can think of when designing a system for the provision of a wireless service. It is a compromise between the demands of different types of applications and the opportunities provided by the current technology.

However, since both application demands and technology are constantly changing, it may be necessary to modify or extend parts of ATM. In this process of reconsideration, it is also worthwhile to consider what modifications would make ATM more suitable for supporting wireless applications. In particular, it may be interesting to investigate the possibility of connections which can be used on a best-effort basis, i.e., without the need for bandwidth reservation and without QoS guarantees.

REFERENCES

[1] Curtis D. Johnson, Wireless Networks Technology, 5th ed, Prentice-Hall International Inc., University of Houston, 1997.

[2] D. Raychaudhuri, L. J. French, R. J. Syracuse, S. K.Biswas, R. Yuan P. Norseman, and C. A. Johnston,"WATMnet: A prototype wireless ATM system for multimedia personal communication," IEEE J. Select. Areas Common, vol. 15, pp. 83-95, Jan. 1997.

[3] Raymond Steele, Lajos Hanzo, Mobile Radio Communication, Wiley 1999

[4] http://

[5] http://www.mks.com

[6] http://cne.gmu.edu/modules/atm/index.html

[7] http://www.doc.ic.ac.uk

[8] http://www.pulsewan.com