COMNET III

Application Notes:

Modeling ATM Networks with COMNET III

Version, 1.0
August 30, 1996

By Matt Falkner

CACI

1. Table of Contents

2. Introduction

Asynchronous Transfer Mode (ATM) is intended to be *the* underlying networking technology of the future. It aims to integrate the entire spectrum of communications traffic, ranging from traditional data traffic applications (file transfers, database request), to increasingly emerging real time applications, such as digitized voice or multimedia applications. It is also designed to be used in networks covering any geographical distance. Whether office LAN, corporate WAN or Internetworks - ATM is supposed to cover it all, thus giving the user the impression of available network services for any type of traffic over any distance.

The result of these ambitious and varying requirements is that the ATM protocol standard has become increasingly complex. Migrating existing networking technologies to ATM or setting up new ATM networks currently challenge any network manager or systems integrator. Furthermore, an additional level of complexity is introduced by the still emerging ATM standards. The ATM Forum has not yet completed its standardization process, so current ATM products on the market run the risk of not fully complying with the standard in the future.

Under such dynamic and uncertain conditions, modeling becomes an essential component for successful network design and management. COMNET III allows you to model your network in order to evaluate the impact of the uncertainties currently associated with ATM. It permits you to build a computer model of your proposed ATM network design under different traffic loads or proposed topologies. It allows you to model the transition from your existing networking technology to ATM, or even to evaluate competing ATM products and their operation within a large ATM network environment.

2.1 Purpose of this Manual

The purpose of this manual is to outline the procedure of modeling ATM networks using COMNET III. It takes the user from a networking perspective to a modeling perspective. The manual is organized into three parts. The first part, chapter three, explains how to model ATM concepts using COMNET III. It focuses on individual technical details such as connection admission control (CAC) or usage parameter control (UPC). The second part, consisting of chapter four, then shows sample ATM networks modeled with COMNET III, thus providing the user with examples of how the modeling concepts of chapter three are combined to make up a model of an ATM network. The last part, the appendices, review the basic concepts of the ATM protocol. It can be viewed as an introduction to ATM concepts for anyone who is not familiar with the protocol. More importantly, however, it provides the user with an account of the underlying technical assumptions which are treated in this manual and which have to be well understood in order to be modeled.

2.2 Underlying Assumptions

A number of assumptions are underlying this manual. First of all, it is not the intent of this manual to serve as a user's or reference manual for COMNET III. The user is assumed to be familiar with the basic concepts of the tool. The individual dialog box options and the functionality of such basic modeling components as nodes or links are not explained here. However, exceptions are made for those concepts in COMNET III which are specific to ATM. Functions and dialog boxes pertaining particularly to ATM are indeed covered. The second assumption relates to the users familiarity with ATM. This manual does not necessarily function as an exhaustive introduction to ATM. Even though the technical aspects of ATM are discussed in some detail in the appendices, the user is still referred to the literature for details. The main purpose of these appendices is to account for the technical details which are covered, and hence also give the user an indication of those details which are not covered. This is particularly important in light of the rapidly changing protocol standards.

To increase the readability of this manual, COMNET III specific keywords are indicated by using a fixed-width font. The usage of the associated building blocks, in particular their entries on the COMNET III menu bar or the palette, can easily be referenced in the COMNET III User's Manual.

3. Modeling ATM Concepts using COMNET III

The third chapter of this manual concentrates on mapping the ATM functions and the underlying theory onto COMNET III. We focus on particular aspects of ATM, such as the modeling of VBR traffic, and show how to set the parameters in COMNET III to model these aspects. Please note that we do not focus on practical performance problems of ATM networks in this part. This is the purpose of the following chapter. Here, we strictly treat individual concepts and map these onto COMNET III.

This chapter is structured as follows: we first of all provide a few general albeit very important remarks about network modeling. These remarks are intended to remind you about the underlying principles of simulation. We then provide an overview of how the ATM concepts map onto COMNET III functions in general. This section is supposed to give you the broad picture on how the ATM functions are represented within the tool. It also covers all those functions which are applicable to more than one of the ATM special cases, for example, those functions which are common to all the different service classes. The third section explains how to model VCCs and VPCs in COMNET III. We then illustrate how to model the different service classes in detail. We continue by outlining the representation of the management functions in a COMNET III ATM model. Finally, we show how to model the individual switching architectures which are described in Appendix A.

3.1 A Few Remarks About Network Modeling

Simulation modeling of ATM networks is a difficult interdisciplinary subject. It requires not only knowledge of the simulation theory itself, but also considerable knowledge about statistics and the underlying ATM theory. The principal idea behind simulation modeling is to represent the topology and functions performed in an ATM network within a model in order to obtain statistical results about the network's performance.

However, it is extremely important to be aware about the distinctions between simulation and emulation. In emulation, the purpose is to mimic the original network or function, and therefore represent every detail. By contrast, in simulation the purpose is to obtain statistical results which describe the operations of the underlying network or function. This implies that in a simulation not every single function has to be represented. You only have to represent those details of the underlying network or architecture which are significant with respect to the statistics which you are trying to obtain from the simulation. From a practical point of view, it is impossible to incorporate every single function which is performed by any of the networking devices in an ATM network and to expect a single computer, often even only a PC, to simulate / emulate such a model.

Admittedly, this description is very vague and does not provide a lot of guidance as to which of the many networking functions should be modeled in a simulation. This is one of the reasons why simulation modeling is a non-trivial task. The key is to incorporate only those details which are significant for answering the problem at hand. This implies that you are aware of the problem at hand, i.e. that you have a clear understanding about what statistics you are looking for. In other words: you need to define a goal for the simulation, and this goal will define the scope of the details which should be modeled. Typical goals which can be modeled with COMNET III are:

- Performance modeling: obtaining utilization statistics for the nodes, buffers, links, ports in the network.
- Resilience analysis: obtaining utilization or delay statistics about the impacts of network device failures.
- Network design: obtaining delay and utilization statistics about alternative network designs and making sure that the design supports the network services requirements.
- Network planning: obtaining performance statistics for future network changes, such as the addition of new users, applications or network devices.

This requirement for network modeling also implies that the same network might result in different simulation models due to differences in these goals. For example, a study might focus on the signaling side of the network. The links might only represent the available bandwidth for signaling. Only the signaling traffic might be represented in such a model. Another study of the same network might focus on the performance under peak traffic loads. In this case, the traffic profiles correspond to the peak loads, the signaling traffic might be omitted in the model altogether.

An even more fundamental question is whether you want to model the internal architecture of an ATM switch or whether you want to model an ATM network. In the first case, you would enter details about the internal architecture of the switch, such as possible bus speeds, buffering characteristics and functions, as well as the interactions between the different components in the switch. If you decide to model an ATM network, you would typically not model the switches at such a low level of detail. The argument here is that the detailed functions performed by the switch do not significantly contribute to the simulation results when looking at an entire ATM network. Adding the detail to model delays in the order of magnitude of nanoseconds - for example by modeling individual processing delays within an ATM switch - is not going to contribute significantly to delay result in the order of magnitude of possibly seconds. The additional accuracy gained from modeling at a high level of detail is far outweighed by the cost and effort required to incorporate the details. What is more, the computer running the simulation must work much harder to keep track of the insignificant events.

The question on what details to incorporate into an ATM simulation is also partly governed by the functions and capabilities of COMNET III. Examining the obtainable results gives an indication what statistics are calculated and hence what details should be included in the model in order to calculate those details. So what results are provided by COMNET III? Basically, three different categories of results are reported:

- Utilization statistics
- Delay statistics
- Error statistics

COMNET III provides detailed reports on the utilization of the different network devices. Nodes, links or buffers for example have capacity parameters (in terms of processing speed or link speed), against which utilization percentages can be calculated. These are reported at various levels of details. Similarly, COMNET III provides numeric results about different networking delay, again at various levels of detail. Message, packet, transmission or setup delays are all included in the reports. With regards to the error statistics, COMNET III records the number of dropped cells, link failures, node failures or even individual disk read / write errors should these be in the model.

By looking at these reports, a number of modeling principles become apparent. Most importantly, COMNET III does not model the contents of individual messages, cells or even frames. Not that this is impossible! The contents are simply not required to compute the above statistics. To compute the delay statistics, only the length of the messages, cells and frames as well as their respective overhead are needed, not the contents in terms of the individual bit values. Similarly, in order to model the concept of a virtual channel within the network, it is not necessary to actually represent network addresses and set their values in the cell header. An easier approach is to take a global view of the network and assume that the names given to the networking devices are globally known. So the routing algorithm is based on the node names and therefore does not have to set or inspect actual network addresses in the cell header. This is an example of the difference between the simulation approach and the emulation approach. Yet another simplification which can be made in simulation modeling relates to the use of statistical functions. Modeling frame or cells errors again does not have to be explicitly done by a link changing a bit in the frame or cell. A statistical function can be used to determine whether the frame or cell has been subjected to noise and is therefore erred. Again, this means that the cell contents do not have to be explicitly represented in order to model a particular function.

These examples illustrate some of the different methods used in simulation modeling which distinguish it from emulation. It is important for you to remember that not every single detailed function in an ATM network has to be modeled and that some functions are replicated by other techniques. After all, the purpose of the simulation is to replicate the functionality of the network operations, not to emulate them.

Table 1 gives you an overview of the ATM modeling capabilities of COMNET III. It lists the main concepts as identified in the previous chapters, as well as whether they can be modeled or not. It also provides you with an indication as to which COMNET III concepts are used to model the respective ATM concepts. Please note that this table only provides a rough overview. The details are explained in the rest of this manual.

ATM Topic	Concept	Modeled in COMNET III?	COMNET III Concepts	Remarks	Section in Part 1	Section in Appendix
Service Classes	CBR rt-VBR nrt-VBR ABR UBR	YES	Session Sources & Transport Protocol / Rate Control		3.3 3.4 3.4 3.5 3.6	A.1
Traffic Contract / Traffic Descriptors	PCR SCR MBS MCR CDVT	YES	Transport Protocol / Rate Control		3.2.2	A.2.1
QoS	Max CTD Mean CTD CDV CLR CET SECBR CMR	YES	Simulation Results		3.2.2	A.2.2
	CER SECBR CMR	NO				A.2.2
Conformance Checking Rules		IMPLIED	Transport Protocol / Policing	Built-into UPC/NPC	3.2.2	A.2.3
B-ISDN Model	Control Plane Traffic	PARTIAL	Message Sources, Triggers,	Functions simulated which are important to compute the available statistics		A.3
	User Plane Traffic	YES	Message & Session Sources			A.3
	Mgmt Plane Traffic	PARTIAL	Message & Session Sources, Buffering Functions	Functions Simulated which are important to compute the available statistics		A.3

Higher Layer Protocols		YES	Transport Protocol, Backbone Network & Transit Networks		3.2.3	A.3
ATM Adaptation Layer		PARTIAL	Transport Protocol / Basic Protocol Parameters & Rate Control	SAR automatically modeled, CS only partially modeled	3.2.2	A.3.1
ATM Layer	Cells	YES	Transport Protocol / Basic Protocol Parameters		3.2.2	A.3.2
	Connection oriented service, PVC / SVC, VP's / VC's	YES	Session Source, Routing Protocol	VP's and VCI's are implicitly modeled	3.2.2	A.1, A.3.2
	Connectionless service	YES	Message Source, Routing Protocol		3.2.2	A.1
	Payloads	IMPLIED	Transport Protocol / Basic Protocol Parameters	not necessary to explicitly model payload	3.2.2	A.3.2
	CLP, Cell Tagging	YES	Transport Protocol / Policing		3.2.2	A.3.2, B.2
	Congestion Notification, Flow Control	YES	Buffer Policies, Available Rate Control	typically only modeled with ABR services, can be modeled using COMNET III's Flow Control algorithms	3.6	A.3.2, B.2
	PTI	YES	Transport Protocol / Basic Protocol Parameters		3.2.2	A.3.2
Physical Layer		PARTIAL	Point-to-Point Links	Physical layer overheads modeled through effective bandwidth	3.2.1	A.3.3

Network Interfaces		IMPLIED	COMNET III Topology, Transport Protocols, Triggers	Interfaces are implied in the topology. Differences can be modeled through triggers and modeling different types of traffic	3.2.1	A.4
Connection Admission Control		PARTIAL		No explicit modeling of resource management and CAC	3.2.2	B.1
	Routing Policies	YES	Backbone Routing, Transit Network Routing		3.2.2	B.1.2
	Address Structure	IMPLIED	Node Names	No explicit addresses used, instead routing based on node names	3.2.2	B.1.2
UPC / NPC		YES	Transport Protocol / Policing		3.2.2	B.2
	Cell Tagging	YES	Transport Protocol / Policing		3.2.2	B.2
	Traffic Shaping	YES	Transport Protocol / Rate Control		3.2.2	B.2
	EFCN, BECN	YES	Transport Protocol / Available Rate Control			B.2
	GCRA	YES		built into COMNET III, used in Rate Control and Policing functions		B.2.1
	Conformance Definitions	YES	Transport Protocol / Policing			B.2.2

Management Functions		PARTIAL	Message Sources	some aspects can be modeled to obtain statistics computed by COMNET III.		B.3
	OAM Cell Flows	YES	Message Sources, Triggers	Cell contents not modeled explicitly		B.3.1
	Performance Management	PARTIAL	Simulation Results	OAM Flow of PM Cells can be incorporated into a model. Simulation results typically report on network performance		B.3.2
	Fault Detection	YES	Triggers			B.3.3.1
	Loopback Control, Continuity Checking	PARTIAL	Message Sources	Only cell flow modeled, explicit functions not modeled		B.3.3.2, B.3.3.3
Switching Architectures		YES	Node Types			A.5
	Crossbar Switch	YES	Switching Node			A.5
	Shared Bus Switch	YES	Router Node			A.5
	Shared Memory Switch	YES	Router Node, C&C Node			A.5
	Delta Switch	YES	C&C Nodes & Subnetwork	model should concentrate on the switching architecture, not network performance for efficiency		A.5

Table 1: Overview of COMNET III’s ATM model building blocks

3.2 Overview of COMNET III's Model Building Blocks

We now outline how COMNET III's building blocks are used to model the ATM concepts. We mainly consider pure ATM networks here, which means that we assume that even the desktops are equipped with ATM devices. This assumption is made to simplify the discussion below. Section 3.2.3 illustrates the relationship of the discussion below with multi-protocol stacks. Please note that this section only provides an overview of the basic modeling components. More details are explained in subsequent sections.

COMNET III generally makes a distinction between modeling the topology and the traffic.

3.2.1 Topology

The topology of an ATM network is typically represented by the node and link building blocks of COMNET III. These are dragged from the palette onto the work area and interconnected to represent the real ATM nodes and their connections.

COMNET III provides 3 different building blocks which can be used for modeling ATM equipment: the `computer and communications node` (`C&C node`), the `router node` and the `switching node`. The `router node` is based on a shared-bus architecture, the `switching node` is based on a cross-connect architecture, whereas the `C&C node` is based on a processor architecture.

In most cases, the COMNET III `router node` would be used to represent the ATM switching equipment, for the simple reason that it is the node building block with the most extensive functionality. Important parameters to look out for here are the `buffer limits` of the switch. These can be specified either by port or for the entire switch. In the case of smaller ATM networks, the internal switching delay may also be significant for the delay statistics, in which case the processing time per packet (i.e. cell) and the bus speed should be specified.

The most important distinction between a COMNET III `router node` and a `switching node` is the blocking behavior. In contrast to the `router node`, the `switching node` models head-of-line blocking. The underlying architecture is based on a switching fabric where all the `input ports` are directly connected to all the `output ports` and switched without any significant delay. Note that if it were not for the difference in blocking behavior, a cross-connect architecture could be represented by the `router node` with zero processing and bus delays.

The COMNET III `C&C node` would be used to represent the ATM workstations in the network. Their primary functionality is to serve as a source or destination of ATM network traffic. Again, the principal parameters here are the `buffer limits` of the workstation, and possibly the processing characteristics for applications.

Note that these nodes represent all ATM devices in the model. Whether a COMNET III node represents an ATM hub, an ATM router or an ATM workstation also depends on how and to which other nodes in the network it is connected, and their respective functions.

To complete the ATM topology in the COMNET III model, `point-to-point links` are used to connect the different ATM nodes. The COMNET III point-to-point building block allows you to set the link speed as well as the framing characteristics of the data link protocol. The link speeds can be entered according to table 24 in Appendix A. Typically, you would enter the effective bandwidth on the link to account for the physical layer framing or signaling overheads. The framing characteristics of the link can be ignored if the ATM protocol is represented in the model as a `transport protocol` (which is strongly recommended). However, you are free to enter the cell minimum and maximum of 53 bytes under the framing characteristics with no overhead bytes to represent the 53-byte cells at this level. Concerning the frame errors, ATM typically assumes a highly reliable physical transmission channel and therefore you can also safely ignore the frame error parameters.

Figure 2 depicts a sample ATM network topology. This topology is based on Figure 28 in Appendix A, but contains additional LANs and an ATM hub. Note that the subnets labeled ‘private ATM’ and ‘public ATM’ contain the respective ATM switches in the backbone.

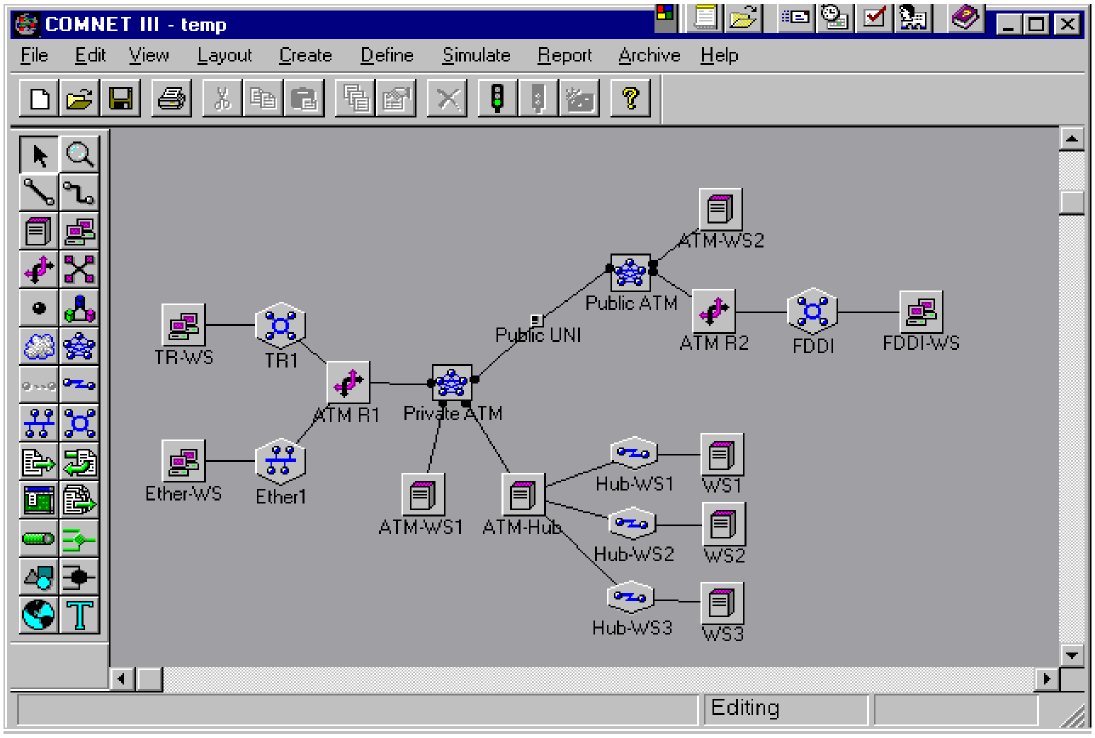


Figure 2: COMNET III ATM topology

3.2.2 Traffic

To generate ATM traffic, you would typically use the traffic sources provided by COMNET III, in particular the session and `message sources`, and connect these to the ATM workstations or end-systems. The `session sources` are used for CBR, rt-VBR, nrt-VBR and ABR traffic, since they represent the connection-oriented traffic types. The `message sources` are used for the connectionless UBR traffic.

These sources provide different types of functions. First of all, they generate messages at the node to which the source is connected. The destination is indicated in the parameter set within the traffic source. Their frequency is determined by the scheduling algorithm as well as the timing details. Perhaps the most important function of the traffic sources from a networking point of view is the association of the message with a `transport protocol`. In COMNET III, ATM is modeled as a `transport protocol` which provides the basic protocol stack. Messages are segmented into packets at the source, and into frames on each link. The frames are re-assembled at the downstream node of the link, the packet is processed and then forwarded through the next link[1].

To model the ATM protocol functions, the details of the `transport protocol` have to be entered. Both the ATM and AAL layers are modeled under the `basic protocol parameters`. These automatically provide the segmentation functions at the source. The destination automatically re-assembles the cells into the original message. The cells are created according to the `data bytes` and the `OH bytes` parameters. Note that COMNET III uses the generic term ‘packet’ for a transport protocol PDU. If the `data bytes` and overhead bytes add up to 53 bytes and the flag `pad to fill packet` is switched on, then COMNET III will always generate 53-byte PDUs, i.e. ‘cells’ in ATM terminology. The overhead parameter should be set to include the AAL overhead bytes as well as the ATM overhead bytes. In the case of AAL3/4, for example, you would enter a packet overhead of 9 bytes, 5 bytes for the ATM header, and 4 bytes for the AAL header. The `data bytes` parameter should then make up the difference between the overhead bytes and the 53-bytes cell size, which in the case of AAL3/4 would be 44 bytes. This implies that a different set of `transport protocol` parameters should be set for each AAL type. The payload type identifier can be modeled using the COMNET III parameter labeled `Protocol ID`, which simply takes a string value. However, this field is only used for modeling different processing delays at the switches, and therefore only necessary if you model at this level of detail.

---

[1] Because the frames are effectively only existing on the link, they are not suitable for modeling ATM cells. The ATM cells are in existence between the source and the destination, and the re-assembly process is only performed at the destination. The intermediary switches do not re-assemble and segment the higher layer PDU.

The CAC functions of the ATM protocol are modeled by using the `session sources` in COMNET III. `Session sources` automatically go through a setup process, where a setup packet / cell is sent through the network from the source to the destination. If the flag `connection oriented routing for sessions` is set under the backbone routing parameters, this setup packet / cell establishes a virtual channel between the source and the destination through which all the data cells will be transmitted.

Two points should be noted here: first of all, the routing algorithm for sessions is only necessary in the case where the networking topology is meshed, or to be precise, where multiple paths exist between any source and its destinations. In the simple case where the sources can only reach the destination through a single path, no routing alternatives have to be evaluated, which eliminates the need for changing the default minimum hop algorithm. Otherwise, the minimum hop routing algorithm should be modified to simulate the dynamic aspects of P-NNI routing.

The second point to note here is that the CAC functions modeled in COMNET III are very limited. Only overbooking methods are modeled, no resource reservation functions are provided. In some special cases, the `session limit` parameters might be applicable to approximate explicit resource reservation functions. When a session setup packet flows through the network, it has to make sure that the `session limit` on both the links and the nodes is not violated. Hence the `session limit` acts as an upper bound for the number of sessions which can be in progress simultaneously at any point in time. If you interpret this limit as representing some sort of resource, then this function may approximate the resource reservation function of ATM, albeit only in those cases where there is no difference in terms of the resource requirements between messages.

The UPC/NPC functions of the ATM protocol are modeled using the `rate control` and `policing` functions of COMNET III's `transport protocol`. The underlying assumption here is that the traffic descriptors are known. The QoS parameters of the traffic contract are part of the simulation output to indicate what quality of service the modeled topology can provide under the given traffic load. The `rate control` function ensures that a cell stream according to a set of traffic descriptors is presented to the network. Alternatively, it can be interpreted to perform the traffic shaping functions. The `policing` function ensures that the cell stream does not violate the specified traffic contract parameters. It is responsible for modeling cell tagging or dropping Typically, there is no need for specifying both `rate control` and `policing` functions at a traffic source. Both functions are based on the GCRA, and the `rate control` function already ensures that the stream is presented according to the traffic contract. The `policing` function might be used independently of the `rate control`, for example at the NNI, or in cases where multiple protocol stacks are modeled.

3.2.3 Other Protocols over ATM

The above discussion assumes that the end-systems are equipped with ATM modules. However, in many cases, you may want to model higher layer protocols which make use of an ATM backbone. This scenario can be modeled in COMNET III using the `transit network` building block. This building block would represent the ATM backbone, to which the other LANs or WANs are connected. Instead of specifying the ATM `transport protocol` parameters at a traffic source, you would specify them at the `transit network` details. The source would take on the parameter set of the higher layer protocols, such as TCP/IP, and thus generate traffic corresponding to this protocol. When the packets are reaching the `transit network`, they are segmented to the `transit network`'s ATM `transport protocol` and re-assembled to the higher layer PDU upon leaving the `transit network`. All the principles about modeling ATM outlined above now simply have to be applied to the `transit network`.

`Transit networks` in COMNET III introduce an additional level of flexibility. They contain the concept of `service classes` and `connection types`. The incoming traffic has a service class requirements measured as an integer. The `transit network` has a list of service classes dividing the integer range into non-overlapping bins. This implies that each incoming packet is mapped onto a single service class. The association with a service class then determines the destinations to which the packet can be transmitted (and hence which types of links that are available to the packet inside the network through the `transit network` routing algorithm). Furthermore, it determines what protocol is used inside the `transit network`. These concepts allow a classification of the traffic and a mapping onto different link speeds and protocol functions. In case you do not need to make such a distinction, simply enter a single service class and a single `connection type` with its respective ATM `transport protocol`.

3.3 Modeling Virtual Channel Connections and Virtual Path Connections

The connection oriented transfer of the CBR, VBR and ABR service classes is modeled using a COMNET III `session source`. Strictly speaking, the model does not distinguish between a virtual path connection (VPC) and a virtual channel connection (VCC). This abstraction of the ATM specification does not need to be modeled in order to obtain the available results. COMNET III therefore focuses on a VCC, and simply ignores that these are bundled into VPC on at least part of the path from the source to the destination.

The COMNET III session can be used to model both permanent virtual circuits (PVCs) as well as switched virtual circuits (SVCs). From a modeling point of view the difference lies in the time the connection is established and its duration. A PVC is typically set up be the network manager and available to the user on a seemingly permanent basis. An SVC is setup upon demand and only available to the user during the transmission phase. To model this difference in COMNET III, you have to make use of the warm-up period of a simulation. A PVC is setup during the warm-up phase of the simulation. Its number of messages times the inter-arrival time should exceed the total simulation run length to ensure that the session never terminates during the simulation run. By contrast, if these conditions are not met, a `session source` automatically models an SVC. Recall that you have to set the flag `connection oriented routing for sessions` under the backbone details to achieve the effect of a logical connection. Otherwise, the `session source` simply generates a series of messages sent using a datagram operation.

The principal question which remains is: how are these VCCs routed? COMNET III supports a routing algorithm at both the backbone level of detail as well as the `transit network`. The choice of routing algorithm depends on the importance of routing to the calculation of the results. In case where the nodes in the ATM network are only connected through single paths, no real routing decision has to be modeled, in which case any of the algorithms suffice. Strictly speaking, however, the algorithm should be set to RIP minimum hop and modified in such a way as to take account of the additional metric which are applied in case of a tie-break. COMNET III does not allow you to base this tie-breaking on the delay or the jitter directly. However, this can be approximated by entering a small `deviation percentage` under the minimum hop backbone details. If multiple shortest paths exist between the source and the destination, COMNET III will effectively balance the load between these if the `deviation percentage` is set. The percentage value should only be about 1%. Consult the COMNET III user's manual for more details on the `deviation percentage`.

3.4 Modeling CBR Services

A fundamental question you should consider when you have CBR traffic in your ATM network is what results you wish to obtain. If your primary focus is on obtaining results for other traffic types, you might be able to simplify the model by simply subtracting the bandwidth reserved for CBR traffic from the available link bandwidth and hence not model the CBR traffic at all! However, if you are concerned with determining the QoS for the CBR connections in light of different traffic loads, or the impact of random CBR connections on other services, then the CBR traffic should certainly be modeled.

The principal characteristics of a CBR service are twofold:

- present the cell stream to the network at a constant rate.
- maintain the constant rate cell stream during the transmission.

CBR traffic is handled by AAL1 to produce a constant cell stream. This is modeled in COMNET III by setting the `basic protocol parameters` and using the `rate control` or `policing` algorithm under the `transport protocol`.

The basic `transport protocol` parameters should be set as follows:

Parameter Name	**Value**
Data Bytes	47
OH bytes	6
Protocol ID	Optional
Error control	Off
Acknowledgments	Off

Table 2: Basic transport protocol parameters for AAL1

Note that the overhead bytes consist of 1 byte AAL overhead and 5 bytes ATM layer overhead.

To format the random cell stream generated by the source into a constant cell stream, you should use the `rate control` function under the `transport protocol`. It is based on a single leaky bucket algorithm which is driven by the PCR and the burst limit parameters. These parameters correspond to the CBR traffic descriptors as described in section A.2.1. The PCR determines the number of kilobits or cells presented to the network per second. The burst limit determines the maximum number of kilobits or cells presented in a single burst. The interval between the bursts is then determined by the underlying GCRA, which will space the bursts such that the PCR is never exceeded. Thus the algorithm generates bursts defined by the burst limit at constant intervals computed by the GCRA such that the PCR is never exceeded. The `rate control` parameters can be summarized as follows[2]:

Parameter Name	**Sample Value**
Constant Info Rate	*151*
Burst limit	*1*
Burst units	*Packets*
Burst type	leaky bucket

Table 3: Rate control parameters for CBR sources

These parameters will produce a constant cell stream of 151 cells per second, each cell sent individually, rather than in a burst of say 10 cells spaced 10 times the interval apart. This is illustrated in Figure 3, which shows the buffer usage of a node with a single CBR source. The traffic load on the source in this case was set to 100000 cells per message. Due to the `rate control` algorithm however, single packet bursts are presented to the network at 1/151 second intervals.

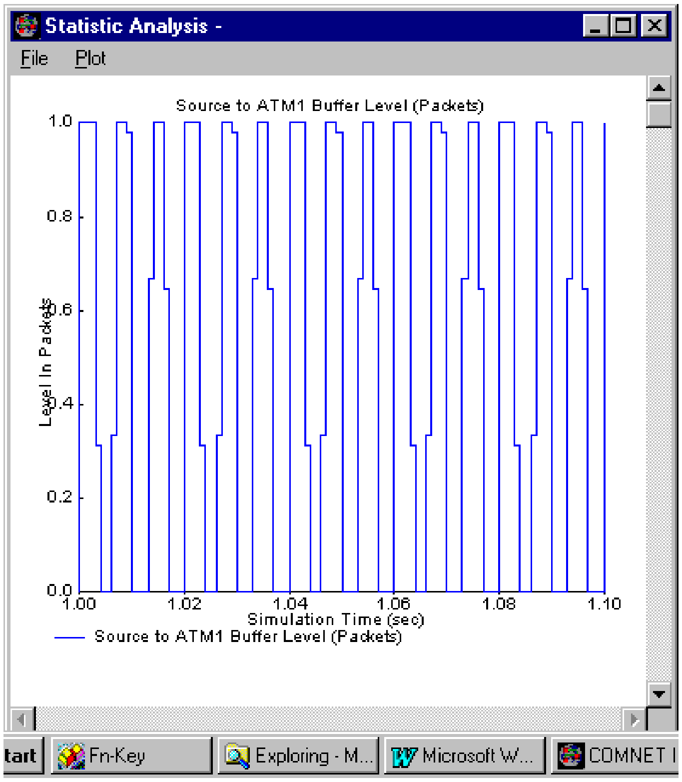


Figure 3: Buffer usage using CBR rate control

---

[2] The text in italics only indicates examples of values. You should replace these with values applicable to your model.

UPC/NPC conformance for the CBR traffic is modeled using the `policing` function of COMNET III. As described in section B.2.2.1 of Appendix B, two versions of this conformance definition are specified, based on the CLP=0+1 stream and on the CLP=0 & CLP=0+1 streams respectively.

The former is modeled by only using one of the two GCRAs provided under the `policing` details. This is done by setting the conformance parameter to GCRA1 only, to set the flag `set CLP` to `never` and to set the `CLP counted` parameter to `CLP=0+1`. This setting implies that cells are either conforming and transmitted, or non-conforming and discarded. No cells are tagged. The rate parameter corresponds to the PCR. The limit burst again looks at the maximum burst which is policed by the GCRA.

The second version is modeled using both GCRAs. The parameter settings in COMNET III depend on whether tagging is supported or not. If it is, then the GCRA1 determines the conformance of the CLP=0+1 cell stream whereas the GCRA2 determines whether the cell should be tagged or not with respect to the CLP=0 cell stream. The `policing` parameters in this case should be set as follows:

Parameter Name	**Sample Value**
Rate 1	*151*
Limit Burst 1	*1*
CLP Count 1	CLP=0+1
Rate 2	*151*
Limit Burst 2	*1*
CLP Count 2	CLP=0
Conformance	GCRA 1 only
Burst units	*Packets*
Set CLP	use algorithm

Table 4: Policing parameters for the CBR CLP=0 cell stream

If tagging is not supported the conformance parameter should be set to `Both GCRA` and the `set CLP` parameter should be set to `never`. This will ensure that both GCRAs are used for conformance checking, albeit on different CLP streams. A cell has to pass both conformance tests to be transmitted, otherwise it is discarded. These settings correspond directly to the specification as outlined in section B.2.2.1.

Note the difference between the `rate control` function and the `policing` function. The `rate control` function maintains the traffic volume and simply adjusts the transmission times and bursts of the message. The `policing` function is more radical by tagging or dropping cells. The two cannot be seen as alternative ways to achieve a constant bit stream.

Both `rate control` and `policing` functions apply to all the generated cells of a particular `session source`, irrespective of whether these relate to a single message instance or to multiple message instances. Hence, if you have overlapping messages within a `session source` (by having a message inter-arrivals shorter than the message delay), then the `rate control` or `policing` functions do not apply only to the individual message instances, but to the entire traffic stream generated by the source. This is one motivation for the `policing` function: ensuring that the aggregate streams conforms to the CBR parameters. If you wish to perform `rate control` on non-overlapping messages, you should split the generation into two or more `session sources`, each of which generate non-overlapping messages.

A further motivation for the `policing` function is to apply it at the NNI at a `transit network`. In this way, a CBR stream generated elsewhere in the model can be policed to verify whether the cells still conform to the traffic contract. The COMNET III results provide an indication on how well the CBR stream was maintained through the network, how many cells have been tagged or even dropped.

Concerning the second characteristic of CBR traffic relating to the maintenance of the CBR cell stream throughout the transmission, COMNET III relies on the priority mechanism to simulate this function. Since CBR traffic has the strictest delay requirements, the priority on the cells should be set to high, and the buffer ranking methods should be left at their default setting based on priority. Recall that COMNET III does not model resource reservations on the nodes and links which could be used. The COMNET III results then provide an indication about the quality of the CBR connection in terms of the transmission delays and their variations. Thus, the QoS parameters of the traffic contract become an output of the simulation.

3.5 Modeling VBR Services

To model both the rt-VBR and nrt-VBR services in COMNET III, the same approach is used as described for the CBR services. A `session source` is used to describe the application data in terms of its inter-arrival time, its size, destination etc. The `transport protocol` is used to cover the AAL and ATM layer functions and to perform traffic shaping and / or UPC/NPC functions.

The differences between rt-VBR and nrt-VBR are modeled in terms of different parameter sets. Nrt-VBR is typically described by a higher MBS than rt-VBR. The COMNET III variable bit rate functions are therefore applicable to both classes.

The `basic protocol parameters` under the `transport protocol` should be set to reflect either AAL3/4 or AAL2, whichever one is used for the traffic stream. Since the details of AAL2 were not yet standardized, we will base the discussion below on AAL3/4. The basic parameter values should be set as follows:

Parameter Name	**Value**
Data Bytes	44
OH bytes	9
Protocol ID	Optional
Error control	default
Acknowledgments	default

Table 5: Basic protocol parameters for AAL3/4

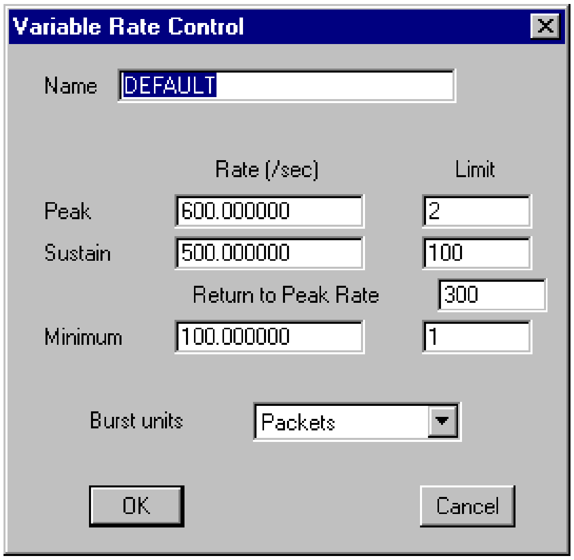
Again, the overhead bytes include the 5 byte ATM layer overhead as well as the 4 byte AAL3/4 overhead. Notice that this only covers the SAR overhead of AAL3/4. The CS header and trailer added to the message (as indicated in Figure 24) are not modeled using the `transport protocol`. In case of large message sizes, this overhead can simply be ignored in the model. In case of small message sizes, it should be added to the message size under the `session source` parameters. Again, the `flow control`, error control and acknowledgment parameters can be ignored for modeling this service.

The `rate control` algorithm again ensures that the generated cell stream obeys the traffic descriptors outlined in section A.2.1. The COMNET III variable rate algorithm should be used. Since VBR traffic is governed by both a PCR and a SCR with their respective CDVT, the function makes use of two GCRAs. The VBR `rate control` function operates as follows:

- The first GCRA allows cells to be generated at the PCR as indicated, using the burst limit. Like above, the inter-arrival time between the bursts is determined by these two parameters.
- The second GCRA monitors the generated stream using the SCR values.
- Eventually, the generated stream will exceed the monitored SCR, in which case the generation is switched to the minimum values.
- The first GCRA now generates cells at the rate indicated under the `minimum` parameter, using its respective burst limit. Again, these two parameters determine the spacing between the bursts.
- Eventually, the monitored SCR will drop below the PCR and approach the MCR.
- If the observed SCR drops below the value indicated under `return to peak rate`, the first GCRA is switched back to the PCR values, and the algorithm starts over again.

Using this function, the cell stream is generated either at the PCR or at the rate indicated under `minimum`, whilst ensuring that the SCR is maintained. Notice that the concept of a minimum cell rate is introduced here artificially. The ATM specification does not call for a MCR as a traffic descriptor. However, since COMNET III does not model the CAC resource reservation functions and the negotiation of the traffic contract parameters, it is used here to drive the variable rate algorithm. The effect of this algorithm is to generate variable bit rate traffic which fluctuates between a minimum cell rate and a peak cell rate whilst maintaining the sustainable cell rate. Rather than stopping the cell generation process for the amount of time required to comply with the SCR again, you are given this additional parameter.

Figure 4 depicts a typical parameter set to generate VBR traffic. Figure 5 shows the `buffer limit` of the generating node and thus the variability in the traffic rate.



Variable Rate Control

Name: DEFAULT

	Rate (/sec)	Limit
Peak	600.000000	2
Sustain	500.000000	100
Return to Peak Rate		300
Minimum	100.000000	1

Burst units: Packets

Figure 4: VBR dialog box

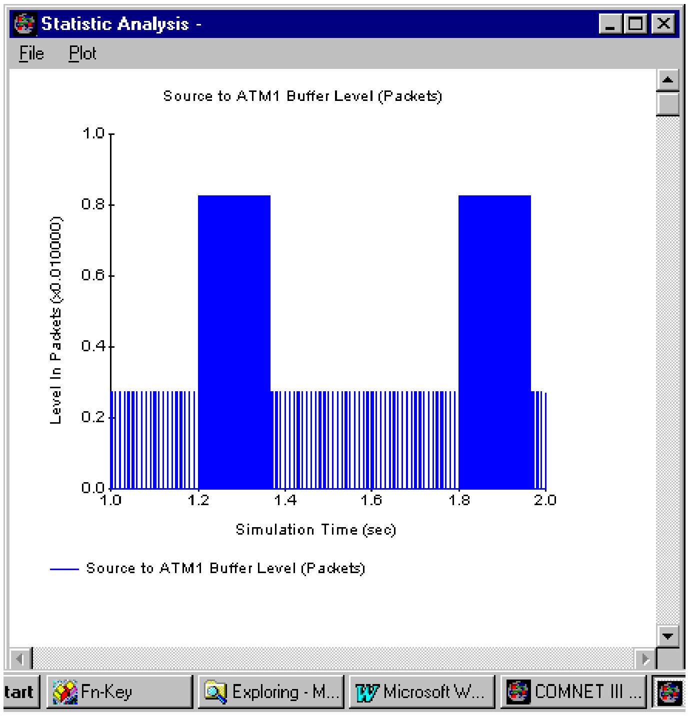


Figure 5: VBR buffer limit.

As illustrated above for the CBR service class, the COMNET III `policing` function of the `transport protocol` is responsible for conformance checking of UPC/NPD, i.e. cell tagging or even dropping. Modeling this function requires knowledge of certain ATM parameters, in particular:

- Does the source transmit a CLP=1 stream or a CLP=0+1 stream?
- What conformance checking rules are used? PCR(CLP=0+1) & SCR(CLP=0) or PCR(CLP=0+1) & SCR(CLP=0+1) - see section 4.2.2.2 for details

If the host transmits CLP=1 cells only, you should set the flag `set CLP` to `always`. Otherwise, the CLP bit and the dropping / tagging operations are determined by the two GCRAs used for variable rate policing. Let us start by mapping the PCR(CLP=0+1) & SCR(CLP=0) cell stream onto the COMNET III parameters.

The settings for the first version in COMNET III again depends on whether tagging is supported or not. If it is, then the GCRA1 determines the conformance of the PCR(CLP=0+1) cell stream whereas the GCRA2 determines whether the cell should be tagged or not with respect to the SCR(CLP=0) cell stream. The `policing` parameters in this case should be set as follows:

Parameter Name	**Sample Value**
Rate 1	*151 (PCR value)*
Limit Burst 1	*1*
CLP Count 1	CLP=0+1
Rate 2	*100 (SCR value)*
Limit Burst 2	*20*
CLP Count 2	CLP=0
Conformance	GCRA 1 only
Burst units	*Packets*
Set CLP	use algorithm

Table 6: Policing parameters for VBR PCR(CLP=0+1) and SCR(CLP=0) cell streams

If tagging is not supported the conformance parameter should be set to `Both GCRA` and the `set CLP` parameter should be set to `never`. This will ensure that both GCRAs are used for conformance checking, albeit on different CLP streams. A cell has to pass both conformance tests to be transmitted, otherwise it is discarded. These settings correspond directly to the specification as outlined in section B.2.2.2.

To map the PCR(CLP=0+1) & SCR(CLP=0+1) cell stream onto the COMNET III `policing` parameters, the following values should be set:

Parameter Name	**Sample Value**
Rate 1	*151 (PCR value)*
Limit Burst 1	*1*
CLP Count 1	CLP=0+1
Rate 2	*100 (SCR value)*
Limit Burst 2	*20*
CLP Count 2	CLP=0+1
Conformance	Both GCRA
Burst units	*Packets*
Set CLP	never

Table 7: Policing parameters for the VBR PCR(CLP=0+1) & SCR(CLP=0+1) cell streams

Recall that this version does not support cell tagging!

The remarks made under the previous section concerning the scope of these parameters with respect to the cell stream carry over. Both `rate control` and `policing` functions consider the entire cell stream generated by the source, regardless of the message instance. To perform these functions on message instances independently, make sure that messages generated by the source are non-overlapping, i.e. that their message delay does not exceed the message inter-arrival time. Similarly, performing both `rate control` and `policing` functions at the same source does not add additional functionality to the model, since the stream generated through `rate control` already conforms and hence the `policing` function should never tag / discard (provided the parameter sets for both `rate control` and `policing` are identical).

The simulation results provide an indication of the QoS which can be expected under the modeled conditions. Again, the priority function should be used in presence of ABR or UBR traffic. Make sure however that the priority does not exceed the CBR priority.

3.6 Modeling ABR Services

To model ABR services requires slightly more complexity within the COMNET III functions than for the VBR and CBR service classes. As explained in sections A.2.1 and B.2.3 below, this class represents ‘best effort’ traffic which is described by a PCR and its CDVT and in addition to this implements a rather complicated congestion control mechanism. The rate at which ABR traffic is presented to the network is primarily limited by the PCR, but may be throttled by the network if insufficient resources are available. COMNET III therefore needs to model this feedback mechanism from the network to the source as well as the throttling mechanism within the COMNET III ABR function.

Like above, the general approach taken by COMNET III to model this type of traffic is to use a `session source` with its typical parameters for inter-arrival times, destinations, message size etc., and to concentrate on the `transport protocol`.

The basic parameters under the `transport protocol` should be set as follows:

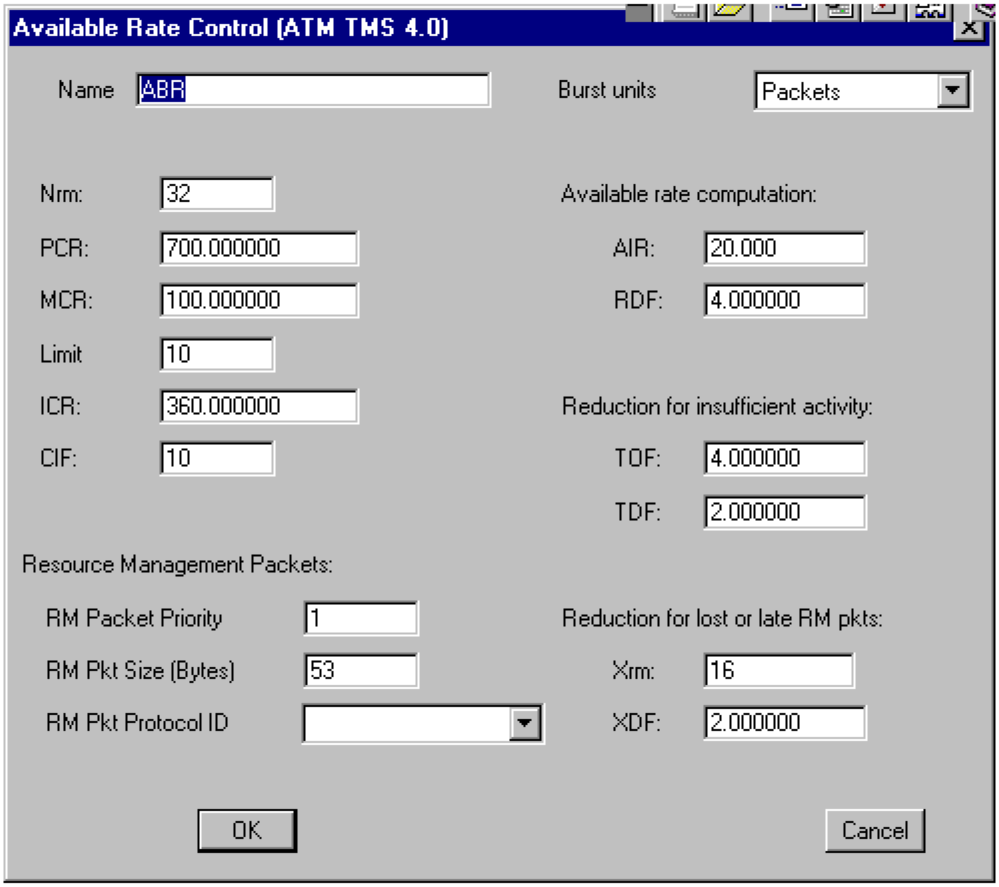
Parameter Name	**Value**
Data Bytes	48
OH bytes	5
Protocol ID	Optional
Error control	default
Acknowledgments	default

Table 8: Basic protocol parameters for AAL5

Note that these values incorporates the AAL5 functions, which in this case cover both CS and SAR functions (cf. Section A.3.1). No simplification has to be made for this AAL in terms of the CS overhead.

The ATM specification also considers AAL3/4 for this service, in which case the discussion on the `basic protocol parameters` outlined in the previous section would apply. Again, no error control, acknowledgments or `flow control` parameters should be set.

The presentation of the ABR stream generated by these parameters is once again governed by the COMNET III `rate control` function, which should be set to `available rate`. The parameters which you are required to enter determine not only the rate, but also the throttling process in case of congestion, as shown in Figure 6. The `rate control` function relies here on the explicit modeling of the feedback, which you are also required to specify at the port buffers in the model. So in order to make the ABR function work in the model, you have to remember to enter both the `rate` `control` parameters as well as to explicitly model the congestion feedback at the port buffers. We will now outline these mechanism in detail.



Available Rate Control (ATM TMS 4.0)

Name ABR

Burst units Packets

Nrm: 32

PCR: 700.000000

MCR: 100.000000

Limit 10

ICR: 360.000000

CIF: 10

Available rate computation:

AIR: 20.000

RDF: 4.000000

Reduction for insufficient activity:

TOF: 4.000000

TDF: 2.000000

Resource Management Packets:

RM Packet Priority 1

RM Pkt Size (Bytes) 53

RM Pkt Protocol ID

Reduction for lost or late RM pkts:

Xrm: 16

XDF: 2.000000

Figure 6: ABR dialog box values

The `rate control` parameters drive the following algorithm:

- Initially, a number of cells are presented to the network according to the initial cell rate (ICR). The size of the first burst is determined by the cells in flight (CIF) parameter. If the ICR takes the same value as the PCR, then you are modeling a rapid start of the transmission, i.e. the source immediately sends according to its peak rate.
- Following this first burst, COMNET III automatically generates a RM cell. This cell is used to determine network congestion.
- If upon return of the RM cell no congestion is indicated, the transmission rate is increased by the additive increase rate (AIR). This will be the available cell rate (ACR). Otherwise, transmission will be decreased by the rate decrease factor (RDF). This is a multiplicative parameter, i.e. the current transmission rate is divided by the RDF to determine the current transmission rate.
- The number of cells per burst subsequently is set according to the `limit` parameter. The spacing of the bursts is again determined by a GCRA according to the ACR.

- Every Nrm user cells, COMNET III automatically inserts another RM cell. Like in the first case, these cells indicate the ACR when they return to the source.
- If no congestion is present, the AIR is applied again. The upper bound for the increase in the transmission rate is the PCR. So when the source is transmitting already at the PCR and a RM cell indicates no congestion, the rate is not altered.
- If the RM cell indicates congestion in the network, the rate at which the cells are presented to the network is throttled again by the RDF.
- The lower limit for the transmission rate is indicated by the parameter `MCR`. The transmission rate of the source will not drop below this limit, irrespective of the congestion in the network.

Two problems might occurs in this algorithm which have to be accounted for:

- If the congestion in the network is so extreme that no RM cells are able to return to the source to indicate the ACR, the Xrm and XDF parameters are applied. The Xrm indicates how many RM cells may be outstanding in the network before the transmission rate is decreased. The amount of the decrease is indicated by the RDF. The Xrm is then automatically decreased by the XDF, indicating that the next rate decrease happens earlier if the network congestion still subsists. Note that this rate decrease is not triggered by an RM cell!
- If the source is not transmitting enough data cells and hence does not send an RM cell for a while, the cell rate is decreased by the TDF*Nrm*TOF. The time period which has to elapse before this decrease is effected is controlled by the parameter TOF*Nrm/ACR. This decrease however is limited by the ICR, such that the current transmission rate never falls below the ICR.

Some of the above parameters have typical values which are already entered in the ABR dialog box in COMNET III. The following table lists these values:

Parameter	**Value**
Nrm	32
RDF	2
TOF	4
TDF	2
Xrm	16
XDF	2

Table 9: Standard ABR parameter values

In order for the RM cells to indicate congestion, the buffers in the network have to be modeled as to mark these cells. This is done in COMNET III by editing the port buffers on the nodes in the network. Under the buffer policy parameters, you can enter directly both the notification type (forward, backward, none) as well as the threshold at which the RM cells are going to indicate the congestion. For example, to model EFCN, you may set the buffer threshold of an output buffer in the network to be 100 cells and the notification policy to `forward`. Whenever a RM passes this buffer and finds more than 100 cells, it automatically indicates the congestion to the source according to the above algorithm. Note that this is critical for modeling the ABR feedback mechanism. If you omit to modify the buffer policy, then the ABR algorithm will never be notified of any congestion and transmit at the peak rate only.

This algorithm corresponds to the description of the ATM ABR algorithm outlined in section B.2.3. Figure 7 illustrates the buffer level of an intermediary switch processing ABR traffic. Note the periodic drop in the buffer level, indicating the threshold value of the buffer, which was set to 200 cells. At this level, the ABR mechanism sets in and the source throttles its transmission, hence the drop in the buffer level.

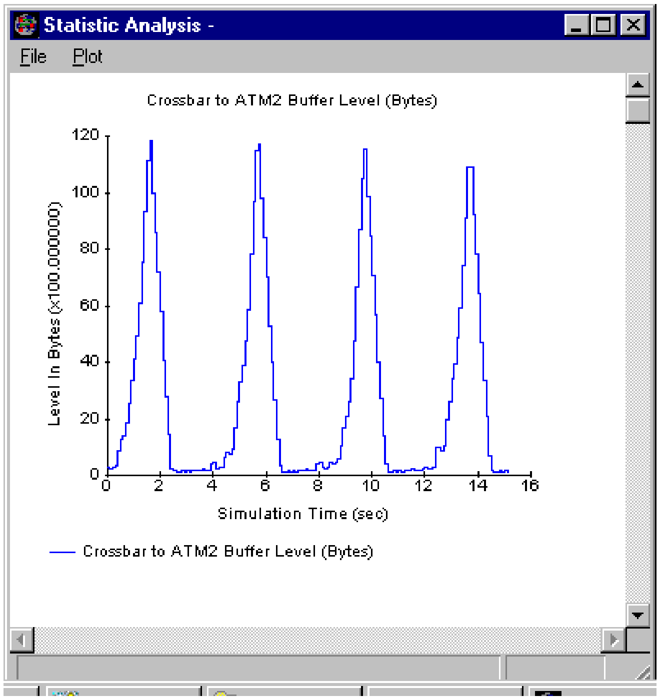


Figure 7: ABR related buffer level

Concerning the UPC/NPC functions of the ABR traffic, the ATM forum provides few guidelines as to how the ABR traffic is policed, as described in section B.2.2.3. The tagging and / or dropping of cells is left up to the device vendors or service providers. The minimal policy indicated is based on a single GCRA and the PCR for the CLP=0 cell stream only. This is simulated according to the following parameters:

Parameter Name	**Sample Value**
Rate 1	*151 (PCR value)*
Limit Burst 1	*1*
CLP Count 1	CLP=0
Rate 2	ignore
Limit Burst 2	ignore
CLP Count 2	ignore
Conformance	GCRA1 only
Burst units	*Packets*
Set CLP	never

Table 10: Policing parameters for the ABR PCR(CLP=0) cell stream

3.7 Modeling UBR Services

As described in section A.2.1, the UBR service class is also considered a ‘best effort’ service from the point of view of the networks guarantees. However, in this case the network makes practically no guarantees about bandwidth or delay and also does not require the user to provide an accurate description of the traffic profile. Cells are transmitted using a connectionless service if spare resources are available, and queued otherwise.

By contrast to the service classes presented above, you would use a COMNET III `message source` to model this kind of traffic. The parameters of a message source determine the inter-arrival time and destination of the message. The `transport protocol` models the different ATM protocol functions.

Since UBR uses either AAL3/4 or AAL5, the `basic protocol parameters` should be set as follows:

Parameter Name	**Value for AAL5**	**Value for AAL5**
Data Bytes	44	48
OH bytes	9	5
Protocol ID	Optional	Optional
Error control	default	default
Acknowledgment	default	default

Table 11: Basic protocol parameters for the UBR service class

No `flow control`, error control or acknowledgment parameters should be set. As outlined in section A.2.1, the ATM specifications only require a PCR and the associated CDVT as a traffic descriptor. This corresponds precisely to the traffic descriptors of the CBR traffic, and hence a similar approach can be taken to model the `rate control`. You should however set the burst rate equal to the PCR to leave the source the flexibility of transmitting as many available packets as indicated in the PCR. Furthermore, the cells should be given the lowest priority to ensure that the other traffic types are given preferential service. In this way, the cells would be presented to the network whenever they are generated whilst still obeying the PCR, but they would only gain access to the networks resources whenever there is spare capacity.

In terms of the UPC/NPC functions for this service class, the modeling depends on whether the service provider or the network actually implements these functions for UBR. In this case, the modeling process of one of the approaches described above carries over.

3.8 Modeling ATM Traffic Management Functions

The traffic management functions outlined in section B.3 can be partially modeled using COMNET III. As mentioned several times above, those functions contributing to the statistics provided by COMNET III can be modeled. Only the cell flows and their frequency are considered here in order to calculate utilization and delay statistics. The cell contents are once again not relevant in the model.

In this case you should be especially critical as to whether these functions have to be modeled at all. In many ATM networks, the network management traffic makes up an insignificant proportion of the total network traffic, and it might therefore be save not to model these ATM services at all. For example, the frequency of FM OAM cells is a function of the number of failures in the network. Failures should be a rare event in your ATM network, so the number of OAM cells generated by such failures should be negligible in comparison to the number of data cells processed during the up-time of the network. Similarly for the PM OAM cell flows. As outlined in section B.3.2, the OAM cells are intended to collect performance statistics particularly for SVCs. In many cases however, the block size determining the number of data cells per OAM cells is large, resulting in only a few OAM cells being generated per SVC.

Since these remarks are not universally applicable for all ATM networks, you might still wish to model these ATM functions with COMNET III. Let us therefore consider the question of modeling the OAM cell flows in detail. You would typically use a separate `session source` between any endpoints of the OAM cell flow to represent the explicitly reserved channels (i.e. PVCs). Recall that COMNET III does not make an explicit distinction between a VCC and a VPC, so you are required to adjust the model to take account of both types of cell flows: for the VCC OAM cells, you would use a `session source` between the endpoints of the VCC connection. Similarly, for the VPC OAM cell flows, you would use a separate `session source` between the endpoints of the VPC connection. The basic `transport protocol` parameters for OAM cells are identical to AAL5. The number of `data bytes` should be set to 48 and the number of overhead bytes should be set to 5. In this case, no `rate control` or `flow control` is specified.

The frequency with which these OAM cells should be generated is model and function dependent. For PM OAM cells, you would typically generate 1 cell for every block of data cells. You could model this by calculating the expected arrival time of such a cell and entering this as the inter-arrival time for the messages under the `session source`. The message size would be set to 1 cell in this case. A simpler approximation would be to adjust the message size of the original data stream by calculating the number of OAM cells which are generated. However, in this case you would not be able to obtain separate statistics for the OAM cell flows in the model, and you would also not model the explicit routing of these cells along the pre-specified VPCs / VCCs.

For the FM functions, a distinction should be made between the continuity checking / loopback OAM cells and the fault detection OAM cells. The former are modeled like the PM OAM cells, using a `session source` and a user-defined inter-arrival time between the endpoints of the OAM cell flow. For the fault detection OAM cells however, you should make use of a `message source`. In this case, the frequency of the OAM cells is determined by the failure characteristics of the nodes and links in the model. COMNET III allows you to make use of triggers to model the interaction between the arrival of an OAM cell and the failure of a topology component. You should set the trigger parameters under the nodes and links in the network. Activating the `Triggers` button on the node or link dialog boxes allows you to select the sources which should be triggered. Under the `Edit` button of these triggers you can then set the triggering rule to be `upon node / link failure`. You should then set the scheduling rule of the respective `message sources` to be `trigger only` to link the generation process of the OAM cell with the failure process. Note that this is the reason for using a `message source`. If you chose a `session source` instead, COMNET III would trigger an SVC upon node / link failure. This is clearly not the case in an ATM network and would hence result in a wrong model.

3.9 Modeling ATM Switch Designs

Much of the discussion above relates to modeling ATM networks. In this section we will briefly outline how COMNET III can be used to evaluate alternative designs for ATM switches. This section relates to the discussion outlined in section A.5 below.

As indicated in section A.2.1 below, COMNET III provides three basic node building blocks to model ATM switches. These building blocks provide functions that allow you to automatically model crossbar, shared bus and shared memory switches. The internal design of your switch can be modeled by adjusting the parameters for these building blocks. The relations between the switch architectures and the COMNET III building blocks is given in Table 7:

ATM Switch Architecture	**COMNET III Building Block**
Crossbar	Switching Node
Shared Bus	Router Node
Shared Memory	Router Node / C&C Node

Table 12: COMNET III building blocks architectures for ATM switches

For example, the `router node` may be used to model different alternative designs for a shared bus ATM switch in terms of

- number of internal buses
- bus speed
- number of ports provided by the switch
- number of processors
- speed of processors

These parameters can be directly altered in the router dialog boxes. Similarly for the other built-in COMNET III node building blocks.

Typically, the COMNET III model would consist of several `C&C nodes` acting as traffic generating nodes. You would connect different CBR, VBR, ABR or UBR sources to these generating `C&C nodes` and connect them through `point-to-point links` to the COMNET III node representing the switch. The traffic would be sent to a destination node at the other end of the switch. In this way, the switches performance could be evaluated under different traffic loads and / or architectures.

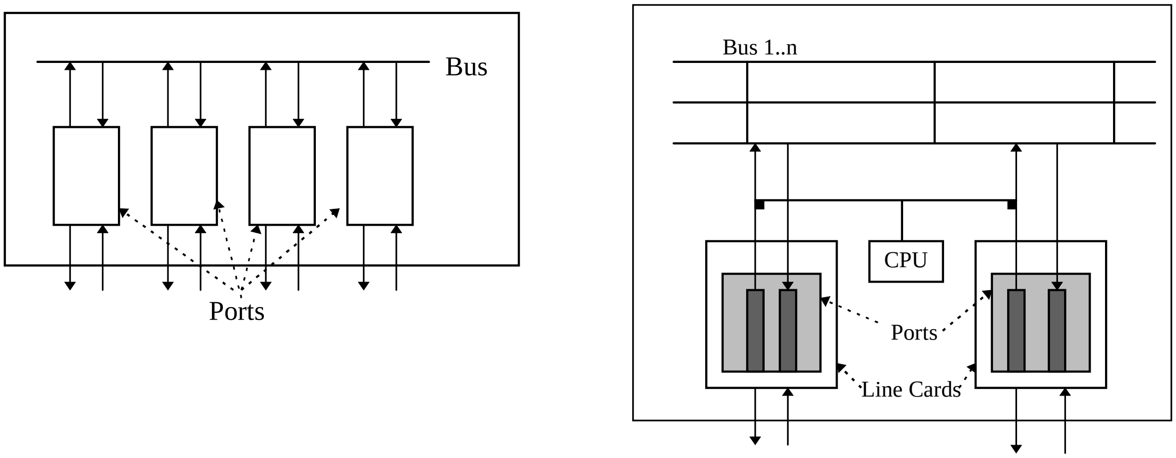


Figure 8: Relationship between conceptual shared bus and COMNET III `router node`.

In the case of a banyan or delta switch you are required to model the internal architecture explicitly using a COMNET III subnet node. This hierarchical building block allows you to place other nodes and links inside to model. In this case, the nodes and links do not represent network components, but internal components of the ATM switch. The internal buffers of the switch would be modeled using `C&C nodes`. The connectivity of the buffers in the switch would be modeled using `point-to-point links`. Figure 9 depicts the architectural view of a banyan / delta switch modeled in COMNET III.

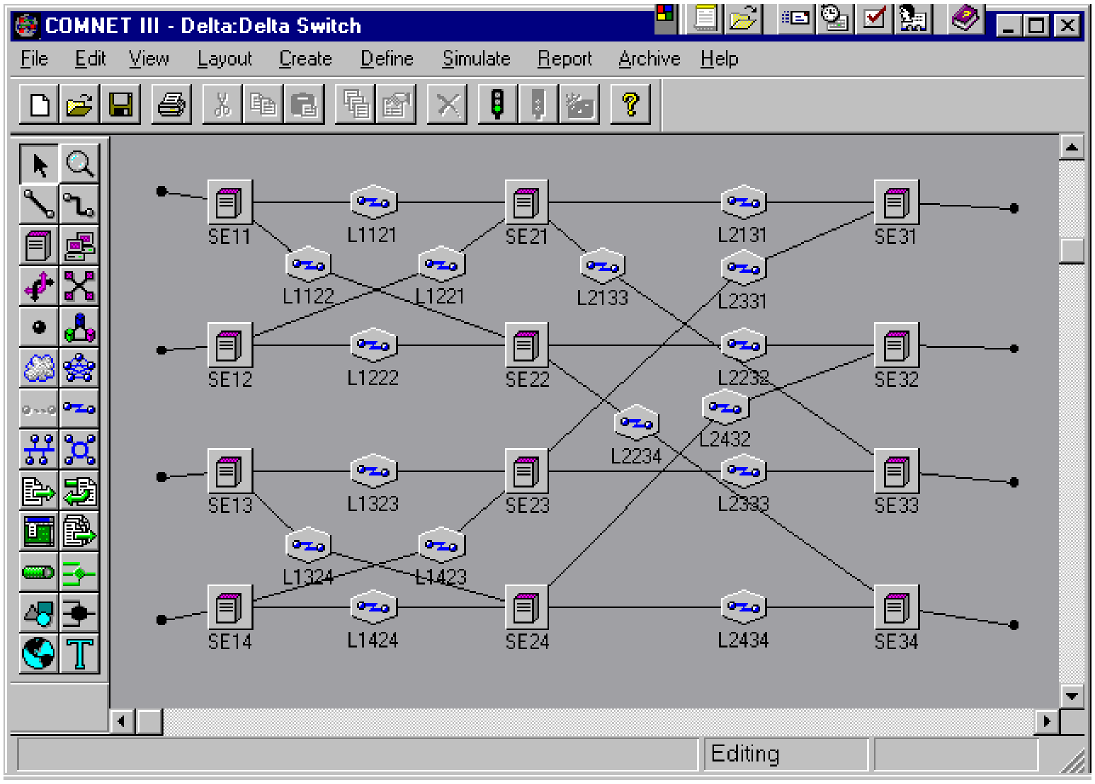


Figure 9: COMNET III model of a Delta / Banyan switch

The `C&C nodes` in this case represent the basic switching elements, which may allow queuing of cells. In this case, you would only use the `buffer limit` parameters and the processing parameters of the `C&C node` to represent the switching element. Depending on how you set the parameters, you may model an input queuing switch, an output queuing switch or both.

Note that this approach can be taken for other switch designs where the pre-defined COMNET III building blocks provide insufficient functions. The building blocks could represent different internal components of a switch. For example, the `C&C node` could represent internal memory if only the disk access parameters are used[3]. Similarly, an internal bus may be modeled using a multi-access link in COMNET III. You may even model more complex interactions of the internal components using `application sources` within the subnet. Applications sources could be used to send messages between the components or model the flow of processes on the different components using the trigger functions.

---

[3] The term ‘disk access’ should be interpreted as memory access. Logically, it represents some memory device, which is typically a hard disk but could also be seen as RAM or a floppy disk device.

4. Example Applications

In the first part of this manual, we outlined how to model the individual concepts of the ATM protocol using COMNET III. We now take the modeling one step further by showing examples on how these concepts are applied to real ATM networks.

Four different examples are given below. The first example examines the problem of analyzing the quality of service for different ATM traffic types. The second example looks at modeling TCP/IP over ATM and the effect on the transmission delay of bringing ATM to the desktop. In the third example, the LANE protocol is modeled. The last example illustrates how COMNET III may be used to model an ATM switching architecture, in particular a delta or banyan switch.

4.1 QoS Evaluation

The network manager of the regional hospital XYZ has recently upgraded the campus backbone to ATM and he is now interested in evaluating the differences between alternative ATM service categories.

For the purpose of this evaluation, the network manager has set up a 128Kbps PVC between the hospital's computer center and the main hospital building. Both locations house 100Mbps Fast Ethernet LANs. The main building contains several 100Mbps LANs which are arranged hierarchically to connect the different wards. The building contains a total of 200 workstations. Each ward has its own LAN and an associated server. The computer center's LAN connects the main server to all the different hospital buildings in order to provide campus wide services.

Figure 10 shows the portion of the hospitals network which the network manager considers relevant for his experiment.

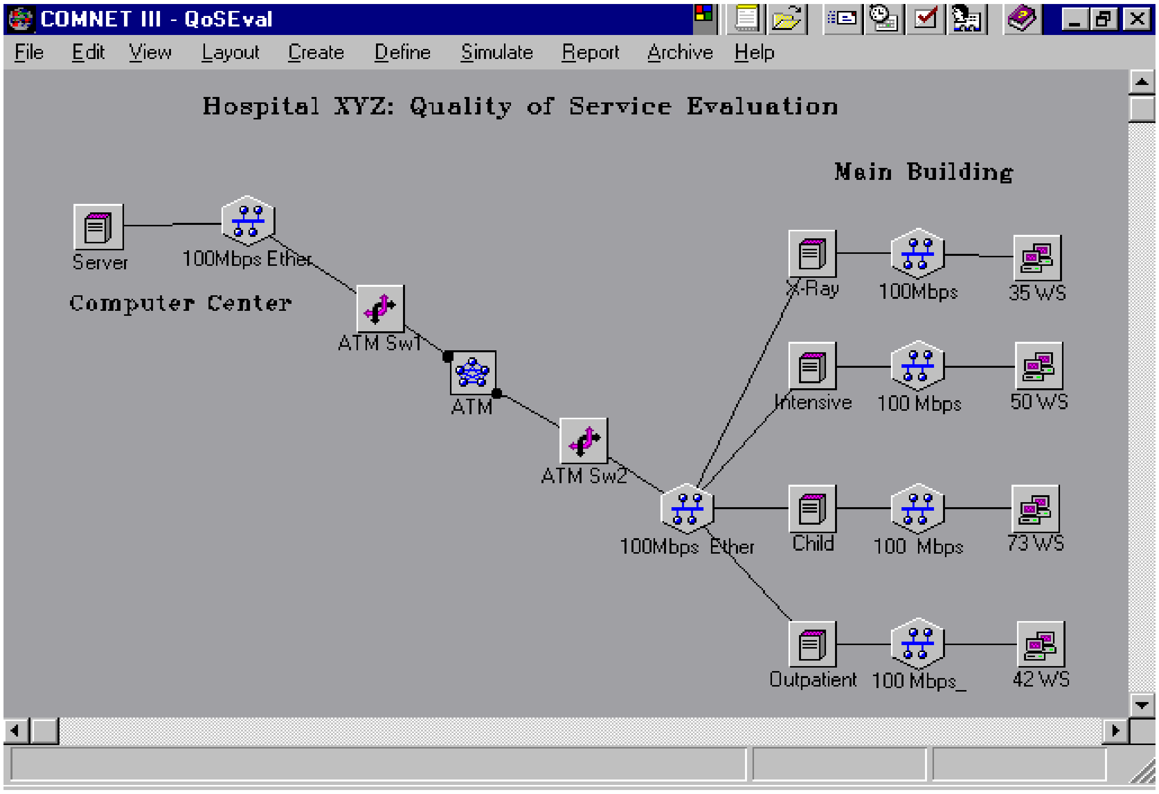


Figure 10: Network topology of hospital XYZ

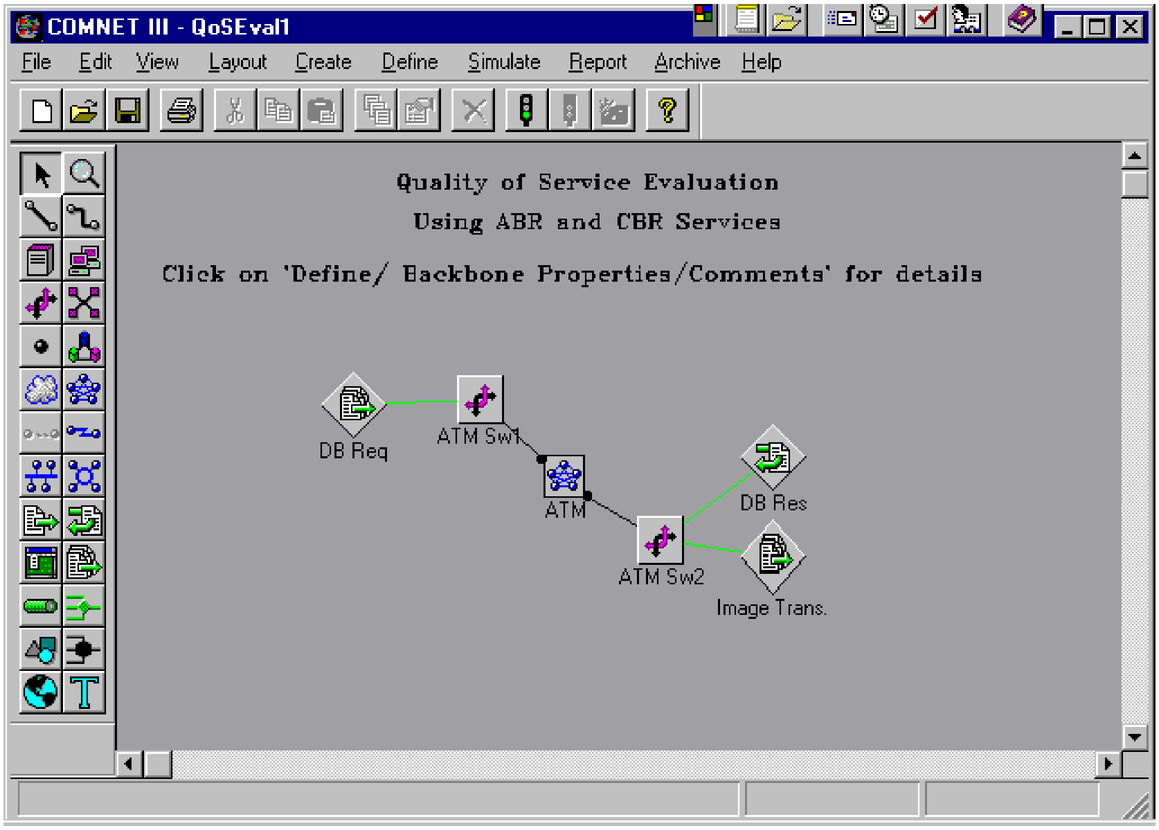
The network traffic is generated by several different applications. Remote traffic between the hospital and the computer center includes database transfers. The hospital's database is located at the computer center to provide campus-wide access of the information. The doctors and nurses periodically query the database to obtain patient information. Another type of remote traffic is generated by the X-ray imaging application. X-ray images are also stored on the server at the computer center, to which they are transferred from the X-ray ward periodically. These images are subsequently available to the doctors on demand. Local traffic on the main building's LAN mainly consists of email, voice mail and local file transfers between the workstations and the local file servers.

The network manager would like to determine the differences in the QoS between alternative ATM service categories. He considers the following alternatives:

- multiplex all remote traffic over the 128Kbps PVC using a VBR service with a SCR of 64Kbps and a PCR of 128Kbps.
- multiplex the image transfer traffic onto a 120Kbps CBR connection and the database traffic onto an ABR connection with a MCR of 8Kbps and a PCR of 20Kbps

The particular QoS parameters of interest are the end-to-end transmission delays. The network manager is particularly worried about the end-to-end transmission delay of the database requests when an image transfer takes place. Long delays for database requests are upsetting for many of the hospital's users. In addition, the image transfers could be scheduled in advance or even be downloaded to the doctor's workstations at night if his or her schedule is known in advance.

The network manager has decided to build two COMNET III models, one for each scenario. He further simplifies the hospital's network topology to only include the ATM backbone in both models. The justification for this simplification is that the LAN's on either side of the ATM backbone are common to both models, such that they could not account for any possible differences in the results. Furthermore, he considers the bandwidth particularly on the main building's LAN to be more than sufficient to handle both the local and the remote traffic. The network bottleneck for the remote traffic is clearly the 128Kbps ATM PVC between the two sites. Any transmission delays or delays resulting from collisions on the LANs are considered insignificant with respect to the end-to-end delay of the remote traffic. The network manager therefore chooses to ignore the modeling of the local LAN traffic. Figure 11 illustrates the modeled topology of the hospital's ATM backbone.



Quality of Service Evaluation

Using ABR and CBR Services

Click on 'Define/ Backbone Properties/Comments' for details

Figure 11: Backbone network model for hospital XYZ

The ATM backbone is modeled using two COMNET III switching nodes and one COMNET III transit net. The switching nodes represent the ATM switches in the hospital's main building and the computer center respectively. The transit net represents the ATM link between the two switches.

The switches in the model are given their default parameter sets. The processing delay of an ATM cell through a switch is also considered insignificant with respect to the total end-to-end delay. Important are the queuing delays and the transmission and propagation delays on the link[4]. The network manager decides to leave the buffer limits on the switches in the model at their infinite values to obtain the maximum buffer utilization as a side-result of the simulation.

The transit net in his model contains only a single point-to-point link with a bandwidth parameter of 128Kbps. This link represents the 128Kbps PVC which has been set up between the two sites. Furthermore, the transit network defines three service classes: ABR, CBR and VBR. Each of these classes is associated with the following minimum and maximum service levels:

	CBR	**VBR**	**ABR**
Min Service Level	1	21	61
Max Service Level	20	40	80

Table 13: Transit network service levels for different ATM classes

Each service class contains a single connection type, labeled CBR, VBR and ABR respectively. This connection type defines the destinations, i.e. the egress switches of the transit network - in this model, the two ATM switches respectively. Furthermore, the connection type also defines the transport protocol used through the ATM backbone. For the CBR service, the transport protocol is defined to be CBR AAL1. For VBR, the transport protocol is defined to be VBR AAL3/4. For ABR, the transport protocol is defined to be ABR AAL3/4. The parameter values for these transport protocols are as follows:

---

[4] See the latest report by S. Bradner of the Harvard Network Devices Test Laboratories.

		CBR AAL1	VBR AAL3/4	ABR AAL3/4
Basic Protocol	Data Bytes	47	44	44
	Overhead Bytes	6	9	9
	Other	default	default	default
Rate Control		constant rate	variable rate	available rate
	Burst Units	Packets	Packets	Packets
	Constant Info Rate	283		
	Burst Limit	1		
	Burst Type	Leaky Bucket		
	Peak / Limit		302 / 1	
	Sustain / Limit		150 / 50	
	Minimum / Limit		50 / 1	
	Return to Peak Rate		70	
	Nrm			32
	PCR			50
	MCR			19
	Limit			100
	ICR			25
	CIF			10
	AIR			5
	RDF			2
	TOF			4
	TDF			2
	Xrm			16
	XDF			2

Table 14: COMNET III parameters for ATM service classes

These service classes on the transit network are associated in the model with the traffic types using the minimum and maximum service level parameters. The network manager thus models the different ATM service classes and their respective traffic contract for the two scenarios he has in mind.

Concerning the traffic in the model, the manager has decided to obtain data for the database requests and the image transfer. Aggregating over the 200 workstations at the hospital's main building, a database request arrived at the ATM switch every 1.5 seconds on average over the measured period. The size of the request and the associated responses were measured to be 1000 bytes respectively. The compressed images were measured to have a size of exactly 1MB. To simplify the traffic side of the model, the manager decides to set up the traffic such that a single image transfer occurs during the simulated period, in addition to the periodic database requests. This does not provide a 100% accurate estimate of the differences between the alternative service classes, but nevertheless a good measure of the order of magnitude.

As shown in the above figure, the network manager models the traffic using COMNET III session sources. The image transfer originates at the hospital side of the model to indicate that the X-ray ward transfers an X-ray image to the computer center. The destination of the session source is set to the remote ATM switch. The image transfer is set to occur 21 seconds into the simulation to allow the database traffic to warm-up the simulated network. The image transfer occurs only once in the simulation. The database traffic occurs every 1.5 seconds between the two sites. The model contains two sources to simulate the interaction between the database request and the database response. The request is given an inter-arrival time of 1.5 seconds, with destination being the remote switch. At the remote switch, the response is triggered by the request message. In both cases, 1000 byte messages are generated in the model.

Under the first scenario, all traffic sources are given a network service level requirement of 30. Since this number lies within the VBR range of supplied network service levels (see above), all traffic is multiplexed onto a VBR service. Under the second scenario, the database traffic is given a network service level requirement of 70 whereas the image transfer is given a network service level requirement of 10. This ensures in the model that the image transfer follows the CBR service class, whereas the database traffic follows the ABR service class.

After running the model under both scenarios, the network manager obtains the following statistics for the end-to-end message delays for the VBR and the CBR / ABR scenarios respectively:

```
                                    QoSEval

                          SESSION SOURCES: MESSAGE DELAY

                     REPLICATION 1 FROM 20.0 TO 320.0 SECONDS

ORIGIN / SESSION SRC:   MESSAGES          MESSAGE DELAY (MILLISECONDS)
  DESTINATION LIST      ASSEMBLED      AVERAGE       STD DEV       MAXIMUM
______________________  _________   ____________  ____________  ____________

ATM Sw1 / src DB Req:
  ATM Sw2                     204         80.728        30.579       483.313
ATM Sw2 / src Image Trans.:
  ATM Sw1                       1     211306.375         0.000    211306.375
```

```
                                    QoSEval1

                          SESSION SOURCES: MESSAGE DELAY

                     REPLICATION 1 FROM 20.0 TO 320.0 SECONDS

ORIGIN / SESSION SRC:   MESSAGES          MESSAGE DELAY (MILLISECONDS)
  DESTINATION LIST      ASSEMBLED      AVERAGE       STD DEV       MAXIMUM
______________________  _________   ____________  ____________  ____________

ATM Sw1 / src DB Req:
  ATM Sw2                     206        591.983      1429.508      5853.411
ATM Sw2 / src Image Trans.:
  ATM Sw1                       1      75204.032         0.000     75204.032
```

Considering the users are highly sensitive to the end-to-end transmission delay of the database requests, the network manager concludes that multiplexing over a single VBR connection is more advantageous. The database requests only take 80 milliseconds on average to be transmitted across the ATM backbone, as compared to an average of almost 600 milliseconds when using the ABR service class. The maximum delay values are even more drastic: in the VBR case, the requests only take 483 milliseconds (less than the average value under the ABR class), whereas the delay is almost 6 seconds. The delays for the image transfer on the other hand are of the opposite magnitudes: under the VBR scenario, the image transfer experiences heavy delays of up to 211 seconds, whereas under the CBR case the image transfer only takes 7.5 seconds to complete.

Upon reflection, the network manager realizes that this result is not surprising. Since he has assigned the image transfer to the CBR service class, the highest priority and most stringent delay requirements are given to this traffic type. The database requests only receive a ‘best effort’ service from the network, and thus suffer when an image transfer is in progress. On the other hand, if both traffic types are multiplexed over the same service class, the backbone does not prefer the image transfer cells over the database request cells, hence the better delay performance.

The side result that the manager wished to obtain on the buffer utilization turned out as follows: the output buffer on the switches showed a maximum utilization of 106 and 159 bytes for the VBR scenario and of 5035 and 8586 bytes for the ABR / CBR scenario. In both cases, these values do not worry the manager since they are well below the switches’ output buffer capacity, even considering the much simplified model.

The above example shows how the ATM features of COMNET III may be used to evaluate different QoS requirements. The network manager could go on to model more details of his network, such as

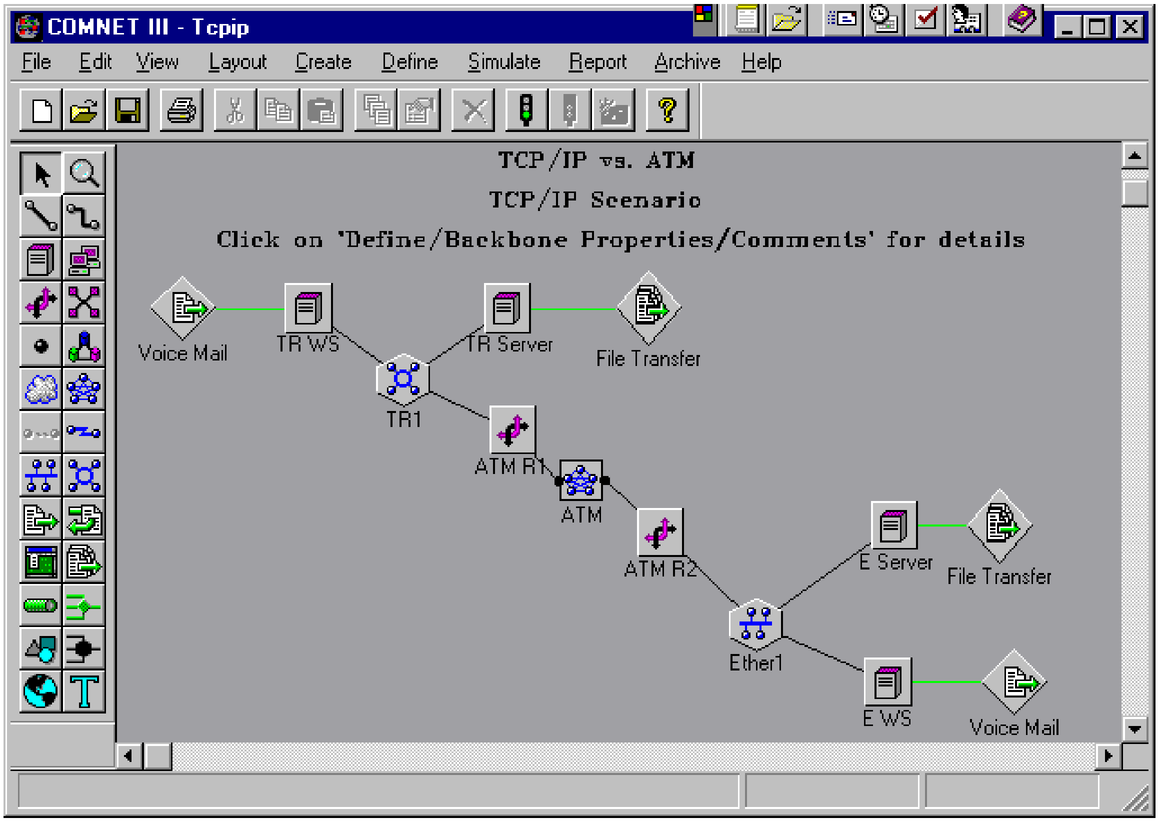
- modeling more image transfers over the backbone
- incorporating statistical message sizes for the database traffic and the image transfers
- extending the model to incorporate the LAN portion of the network
- considering alternative bandwidths for the PVC.

Furthermore, he could further study the parameter settings for the different service classes to determine how these affect the QoS of the traffic. Examples here are to experiment with the PCR or SCR of the traffic contract, hence evaluating different traffic shaping functions or settings for the traffic contract.

4.2 TCP/IP vs. ATM at the Desktop

This example considers a small high-tech company with two offices located on the east coast and the west coast respectively. The network manager of the company realizes the advantages offered by ATM, in particular concerning the provision of real-time services between the offices such as video conferencing. This would save considerable travel costs of employees between the two locations. She has already connected both offices using an ATM service from a service provider. However, the network manager now intends to complete the process of upgrading to ATM by replacing the existing LAN technology in each office by ATM. In order to justify her plans with the management, she decides to model both scenarios to show the benefits of bringing ATM to the desktop.

The company's network topology is currently configured as follows: The east coast LAN consists of a 10-Base-T Ethernet LAN to which the offices workstations and the local server are connected. The west coast office has a similar setup, except that the LAN is a 16Mbps Token Ring. Each office already has a small ATM router which is also hooked up to the LAN. To connect both offices, the company has signed up with a service provider and leased a 64Kbps ATM PVC. The access links to the service provider's ATM network in both offices therefore also run at 64Kbps. The network topology is illustrated in Figure 12.



TCP/IP vs. ATM

TCP/IP Scenario

Click on 'Define/Backbone Properties/Comments' for details

Figure 12: COMNET III model for the small high tech company

The network traffic is both local and remote. The local traffic is made up of running the company's proprietary client-server applications off the local server. Office applications (word processors, spreadsheets) are also installed on the local servers. The remote traffic types consist of mainly file transfers, in particular exchanging proprietary data files, as well as voice-mail between offices. Remote traffic is based on the TCP/IP protocol stack. The voice-mail application is transmitted across the service provider's ATM network using an ABR service. The file transfer across the ATM network uses an CBR service, since the data is considered delay sensitive. The protocol stack currently maintained for the remote applications is as follows:

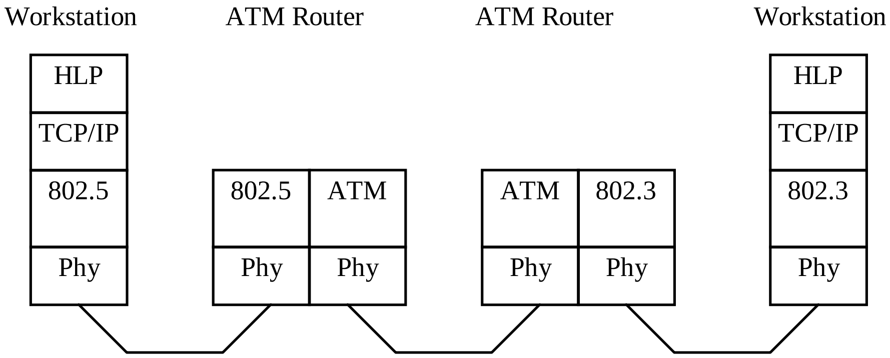


Figure 13: TCP/IP over ATM Protocol Stack

The network managers proposal includes upgrading the current workstations to include ATM network interface cards. The workstations would then be directly connected to the ATM router through 1.5Mbps DS-1 links, thus replacing the existing LANs in each office.

In order to support her case against management, the network manager has decided to build two simple example models illustrating the proposed upgrade. She has decided to include only the following components in her model to allow her to effectively argue her case:

- The local office server
- A representative workstation
- The LANs in each office with reduced bandwidth or the point-to-point connections of the workstations in the proposed case
- The ATM routers in each office
- The service provider's ATM connection between both offices
- The remote voice mail and file transfer applications

Note that she has decided to reduce the available bandwidth of the office LANs to 4Mbps. She wants to argue that this is the minimum available bandwidth under the existing case. This simplification allows her not to explicitly model the local LAN traffic. Furthermore, she has decided to model the 64Kbps leased PVC in both cases to allow a comparison between the results. She realizes that this link will have to be upgraded when fully supporting ATM real time services, but for the simplicity of her argument against management, she is only going to model the existing network and compare it with model illustrating ATM to the desktop, all else being equal. She already considers building a third model to illustrate the upgrade of even the inter-office link to DS-3 speeds. The model for the existing network is depicted above in Figure 12.

The servers and the representative workstation in each office are all modeled as COMNET III Computer & Communication nodes with their default values. The manager is not interested in the local processing statistics for her argument. The routers in the offices are modeled using the Novell Multi-protocol Router Plus, V2.1 and the IBM 6611, V1.1 parameter sets in COMNET III. She found those values to be representative of the ATM routers currently installed in each office. The LANs in each office are modeled as COMNET III token ring link and Ethernet link respectively. The token ring link is given the 802.5 16Mb Token ring parameter set with a modified bandwidth of 4Mbps. Similarly, the Ethernet is given the 802.3 10 Base T parameter set with a modified bandwidth of 4Mbps. The routers are connected to a transit net in the model, using a 64Kbps point-to-point link. This link is modeled inside the transit net to represent the capacity the company has currently leased from its service provider. The transit net allows the manager to model the specific protocol stack of TCP/IP over ATM. It is given two service classes: CBR and ABR. Each service class contains a single connection type, specifying the egress router and the respective ATM protocol respectively. The parameter settings for the transport protocol of these connection types are identical to those outlined in the previous example.

The model also includes 4 traffic sources to represent the remote traffic in the network. The file transfer traffic originates at both servers. The destination in each case is the remote workstation. The size of the file transfer is set to 1Mb, occurring exactly once in the simulation. The file transfer is given the transport protocol ‘TCP/IP’ in COMNET III and a network service level of 10. With these settings, the file transfer of 1Mb will be segmented into TCP/IP packets across the local LAN. Upon arrival at the ATM routers, these TCP/IP packets will then be further segmented into ATM cells following the CBR service class (i.e. using AAL1). The manager sets the priority of this traffic source to 80 to indicate the higher priority of CBR traffic. The voice mail traffic occurs periodically throughout the simulation. The given inter-arrival time is 10 seconds exponentially distributed. This value reflects the frequency with which all the workstations in each office send voice mail to the remote office. The size of the messages is set to 10Kb, again following the TCP/IP transport protocol. In this case, the net service level parameter is given the value 70 to indicate that the voice mail packets are transmitted across the ATM PVC using the ABR service class. Again, this traffic is modeled using a session source to reflect the connection-oriented nature of ATM. All traffic in the model is symmetric between the two offices.

The second model in support of her case against management represents the proposed network using ATM to the desktop. The network manager here simply replaced the LANs in both office locations by two point-to-point links with parameter sets DS-1. The values of this parameter set are as follows:

Parameter Name	**Value**
Number of Circuits	1
Bandwidth / circuit (from node X)	1536
Bandwidth / circuit (from node Y)	1536
Framing characteristics	all values at 0

Table 15: COMNET III parameters for a DS-1 link

Furthermore, the manager has also changed to transport protocol of all traffic sources to ATM. The file transfer sources are given the CBR AAL1 parameter set with the following values:

		CBR AAL1
Basic Protocol	Data Bytes	47
	Overhead Bytes	6
	Other	default
Rate Control		constant rate
	Burst Units	Packets
	Constant Info Rate	283
	Burst Limit	1
	Burst Type	Leaky Bucket

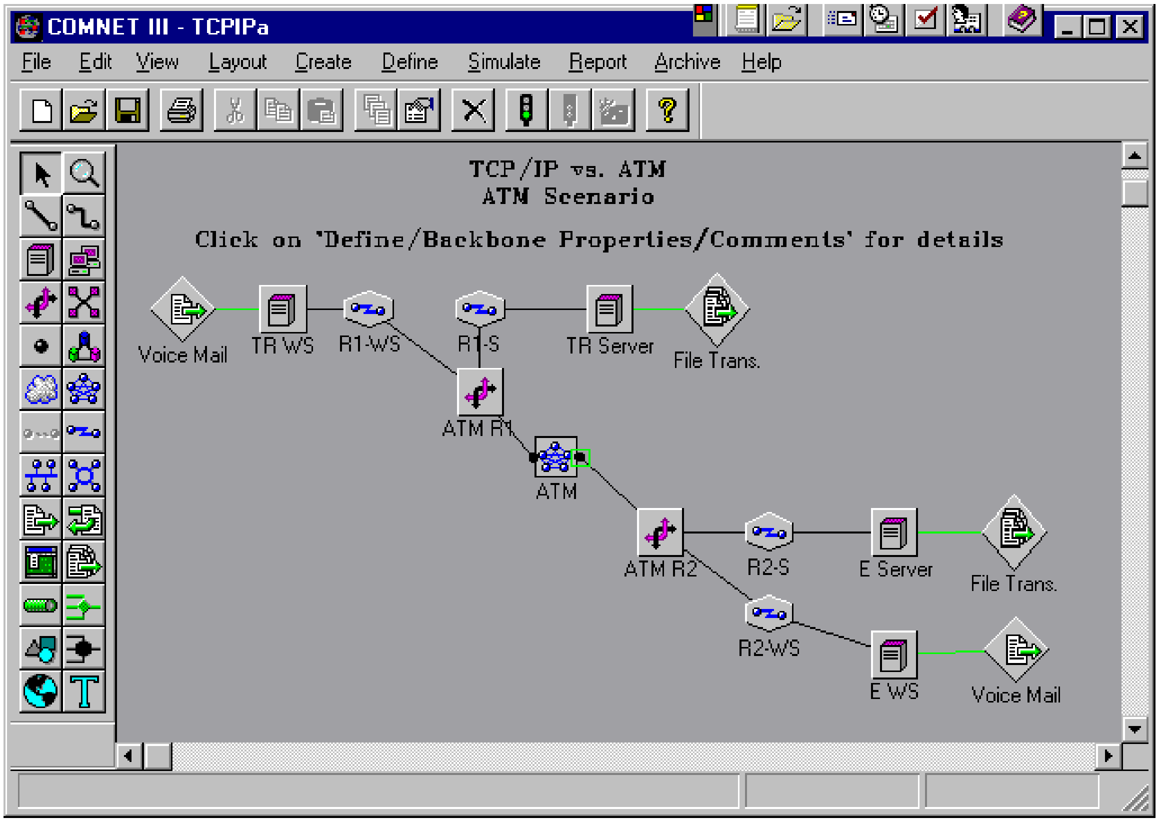
Table 16: COMNET III parameters for the CBR AAL1 service class

The voice mail traffic is given the ABR AAL3/4 parameter set with these values:

		ABR AAL3/4
Basic Protocol	Data Bytes	44
	Overhead Bytes	9
	Other	default
Rate Control		available rate
	Burst Units	Packets
	Nrm	32
	PCR	50
	MCR	19
	Limit	100
	ICR	25
	CIF	10
	AIR	5
	RDF	2
	TOF	4
	TDF	2
	Xrm	16
	XDF	2

Table 17: COMNET III parameters for the ABR AAL3/4 service class

With these settings, the alternative model represents a pure ATM protocol stack, where the traffic is already segmented into ATM cells at the source. For this reason, the framing characteristics on the point-to-point link in the model have been set to 0. The topology of her second model is depicted in Figure 14.



TCP/IP vs. ATM
ATM Scenario

Click on 'Define/Backbone Properties/Comments' for details

Figure 14: Alternative COMNET III model for the small high tech company

After running the simulation for both models, the network manager realizes that the end-to-end transfer delays differ significantly. The reported delay for the file transfer in the TCP/IP case is 205 seconds, whereas the delay in the ATM case is only 140 seconds. This supports the argument that the manager was hoping to make. Another interesting result she finds in the COMNET III reports relates to the end-to-end packet / cell delay. In the existing case, the delay per TCP/IP packet is 0.5 seconds on average and 2 seconds at the maximum. By bringing ATM to the desktop, the end-to-end delay for a single cell is 35ms with a maximum of 60ms. She considers using these values to support her argument of ATM being more suitable for real-time applications and less sensitive to individual cell losses. Both simulations confirm that under the modeled cases, no cell or packet losses occur.

As an enhancement to her model, the network manager considers modeling more appropriate links speeds for ATM, in particular when the ATM connection between the two offices is upgraded to at least a DS-1 or even a DS-3 link.

4.3 Modeling ATM LANE Networks

This example illustrates how COMNET III can be used to model the LANE protocol. The ATM Forum intends to allow existing applications designed for traditional LANs to use ATM by specifying a LAN emulation protocol (LANE). The application assumes that the underlying networking technology is still based on one of the traditional LAN protocols. However, these are only emulated here through LANE. One of the principal functions is therefore to perform the address resolution between the MAC addresses and the ATM addresses.

In LANE, a number of components are defined:

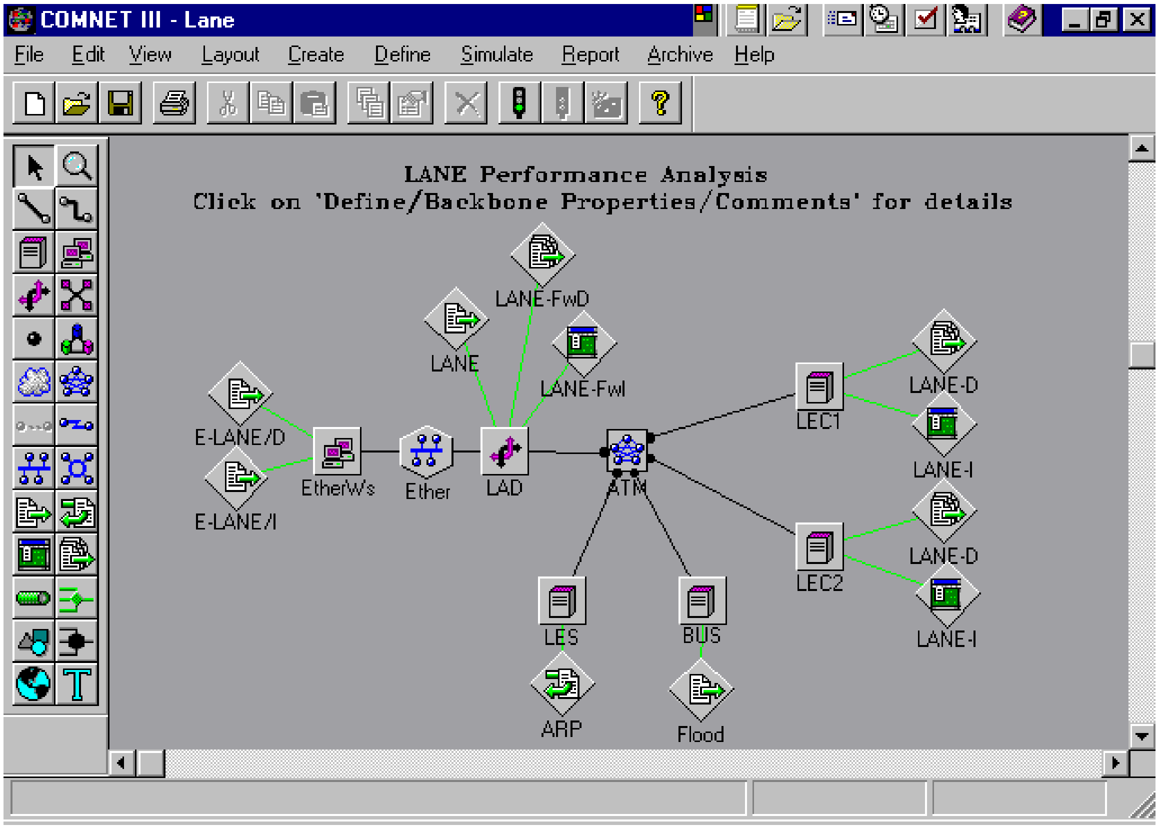
- LAN Emulation Client (LEC)
- LAN Emulation Server (LES)
- Broadcast and Unknown Server (BUS)
- LAN Access Device (LAD)

Together, all these component allow the formation of a virtual LAN (VLAN). The interactions between these VLAN components is as follows: upon receipt of a traditional LAN frame from the higher layer protocols, the LEC performs its address resolution function. Each LEC maintains a mapping / cache of known MAC addresses to ATM addresses. However, this mapping might not be exhaustive and furthermore is purged periodically. In case when a frame is received with an unknown MAC / ATM address, the LEC sends an address resolution request to the LES. Each member of the VLAN has to register its MAC / ATM address with the LES upon joining the VLAN. If the LES has an entry for the required ATM address, it replies to the LEC, which is then able to set up a direct connection to the destination using the connectionless AAL5 service.

In some cases however, even the LES might not have an entry for the MAC / ATM address pair, in particular if the destination MAC is hidden behind a LAD or if it is part of another VLAN. For this reason, the LEC also sends the first few frames of the transmission to the BUS, which floods these frames to all the members in the VLAN. At the same time, the LES initiates an address resolution protocol, possibly prompting other LESs for the requested address. Should one of these flooded frames reach the destination, possibly through a LAD, the destination informs the LES about its address. Again, once the address resolution is completed, the source LEC establishes a direct ATM connection to the destination using AAL5.

These complex interactions between the different modules in the VLAN can be modeled using COMNET III. Figure 15 illustrates a sample LANE model. The assumptions underlying this model are as follows:

- Only the members of the VLAN are included in the model.
- Each module (LEC, LES, BUS) is modeled on a separate physical node.
- The explicit address resolution protocol of the server is not modeled. Instead, the process is modeled by a time delay of 0.5 seconds. This simplification ignores some of the network traffic which is generated by the ARP process. However, this traffic is considered insignificant with respect to the other traffic load in the network.
- The LEC is assumed to have the destination ATM addresses cached in 50% of all transmissions. The other 50% of the transmissions have to go through the LANE process as outlined above, simulating the cases where the cache entry has either been purged or when the destination is hidden behind a LAD.
- In the case where the LEC has to request the address from the LES, only a single packet is sent to the BUS for flooding.
- Only VLAN traffic is modeled.



LANE Performance Analysis
Click on 'Define/Backbone Properties/Comments' for details

Figure 15: COMNET III LANE model

The model topology consists of a number of nodes and links. The nodes ‘LEC1’ and ‘LEC2’ represent LECs which are directly part of the VLAN. They are assumed to have a LANE module in their respective protocol stack. The node labeled ‘LES’ represents the LES module. This is assumed here to be implemented on a separate server. Similarly, the node labeled ‘BUS’ represents the BUS explicitly in the model. Both nodes perform only the functions of the LES and the BUS respectively. The node named ‘LAD’ represents a LAD, which is connected to both the VLAN as well as a traditional Ethernet LAN. All these nodes are modeled using a COMNET III Computer & Communications (C&C) node. They are connected to the VLAN through point-to-point DS-1 links. Apart from the LAD, 10 workstations are also connected to the Ethernet LAN, using a COMNET III Computer Group. All C&C nodes retain their default values. Similarly, the Computer Group also retains its default parameter values with the exception of the ‘number in group’ parameter, which has been set to 10. The VLAN itself is modeled by a COMNET III transit net. A single ATM switch represents the physical aspects of the VLAN in the model. All the nodes on the VLAN are connected to this switch through the DS-1 links. The transit net retains it default network service class and its default connection type. However, the protocol of this connection is set to ATM AAL5 with the following parameter values:

		ATM AAL5
Basic Protocol	Data Bytes	48
	Overhead Bytes	5
	Other Parameters	default
Flow ctrl		none
Policing		none
Rate Control		none

Table 18: COMNET III parameters for AAL5

A number of COMNET III traffic sources define the VLAN traffic. The LECs generate direct and indirect traffic, the former representing those frames where the LEC has the destination address cached, the latter representing the case where the LEC has to go through the LES. The direct traffic is generated using a COMNET III session source. The source generates traffic with an inter-arrival time of 10 seconds, exponentially distributed. The size of the messages is 1500 bytes representing TCP/IP packets. An average of 10 TCP/IP packets is generated per session, normally distributed with a standard deviation of 1 packet. The session thus generates a 15000 bytes LANE traffic burst. Each packet follows the transport protocol labeled ‘Ethernet’. This protocol basically represents the Ethernet frame which is generated by the application. The parameter values for this protocol have simply been copied over from the COMNET III 10 Base T link parameter set. The destination of this direct message source is either the remaining LEC, or the LAD. With these settings, the following protocol stack is implemented:

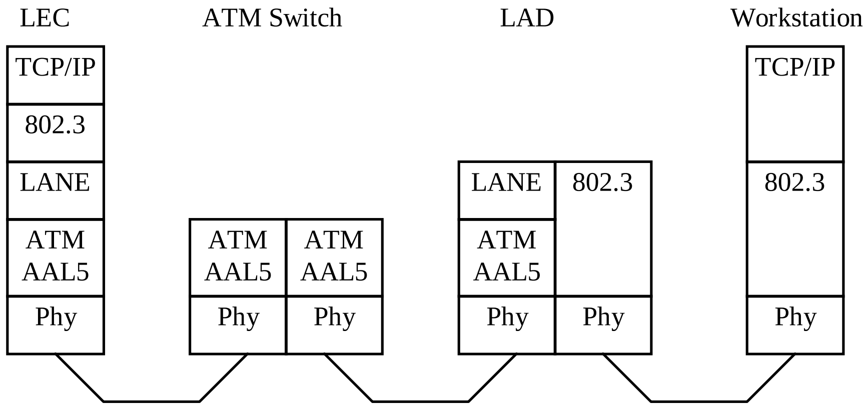


Figure 16: LANE protocol stack

Notice that the figure also illustrates a transmission between a LEC and a workstation on the traditional Ethernet LAN. The operation of the LAD is explained below.

A COMNET III application source is also connected to each LEC. This source models the indirect traffic which has to go through the LES. Every 10 seconds distributed exponentially, the application executes the following command sequence:

- Send ARP
- Broadcast
- Await Response
- LANE

The first command sends a 30 byte message to the LES to simulate the address resolution request. This command is defined as a global transport command, and hence it is available to all the application sources in the model. The request again follows the Ethernet protocol. Its frames are segmented into ATM cells within the VLAN. At the same time, the application sends a message to the BUS by executing the command ‘Broadcast’. This is a local transport command with message size 1500 bytes, the equivalent of 1 TCP/IP packet. Again, the packet is sent using an Ethernet frame, which is segmented into ATM cells on the VLAN. After these two transmissions, the application then waits for a response from the LES. This is implemented in the model in form of a global filter command. The message to wait for is set to ‘ARP’, which corresponds to the message text by which the LES responds to the request. Upon arrival of such a message, indicating that the address resolution request has been terminated, the LEC then establishes a connection with either the other LEC or the LAD. This is modeled using a local session command, the parameters of which are identical to those outlined under the direct transmission above.

Connected to the LES is a single COMNET III response source. This source is triggered upon arrival of the 30-byte message which is generated by the execution of the command ‘send ARP’ on the LECs. A time delay of 0.5 seconds then elapses to simulate the address resolution process. This is a simplification made in this model, since lookup of the addresses in the LES or the address resolution requests to other LESs are not explicitly represented in the model. After the delay, the response source returns a 30-byte message to the requesting LEC, again using a single Ethernet frame. This message is associated with the text ‘ARP’, which upon arrival at the LEC triggers the continuation of the LEC’s command sequence.

A single COMNET III message source named ‘Flood’ is attached to the BUS. This source is responsible for flooding a single TCP/IP packet. The source is triggered by the message ‘Broadcast’, which is sent by executing the command ‘Broadcast’ on the LEC. Upon arrival of such a message, the source then forwards the incoming message to all the members on the VLAN, using Ethernet frames. The parameter responsible for the calculation of the message size, ‘Msg size calc’, is set to replicate the size of the incoming message by using the ‘A * Msg size +B’ algorithm with the parameters 1 and 0 for A and B respectively. In this way, the model can easily be modified to simulate the flooding of several packets, not just a single packet as indicated in the list of assumptions. To determine the destination of this flooded message, the ‘multicast’ algorithm is used. The list contains all the nodes which are part of the VLAN.

The interactions of the VLAN with the traditional Ethernet LAN are modeled at the LAD. This node is associated with three COMNET III traffic sources labeled ‘LANE’, ‘LANE-FwI’ and ‘LANE-FwD’. The principal function of these sources is to convert the protocol stack between the Ethernet LAN and the VLAN. The two sources ‘LANE-FwI’ and ‘LANE-FwD’ handle the traffic from Ethernet to the VLAN. The source ‘LANE’ handles the traffic in the other direction.

The message source ‘LANE’ is triggered by indirect and direct messages from the LECs respectively. Notice that the LECs operations as described above do not establish a connection with the workstations on the traditional LAN directly. Instead, the direct and indirect traffic is transmitted to the LAD, since the connection-oriented ATM protocol cannot be part of the traditional Ethernet LAN. The source ‘LANE’ is triggered by any message originating from either LECs, then assembles the message and transmits it in connectionless mode to one of the workstations on the traditional LAN. This protocol conversion is indicated in Figure 16 above. The parameter settings for this traffic source are as follows:

Parameter Name	Value	Additional Values
Schedule by	Received Message	‘LANE’
Rec msg delay	none	
Msg size calc	A *Msg bytes + B	
A	1	
B	0	
Msg text option	Copy message name	
Priority	1	
Routing Class	Standard	
Transport Protocol	Ethernet	
Net svc level	1	
Dest type	Random list	EtherWS

Table 19: COMNET III parameters for the ‘LANE’ message source

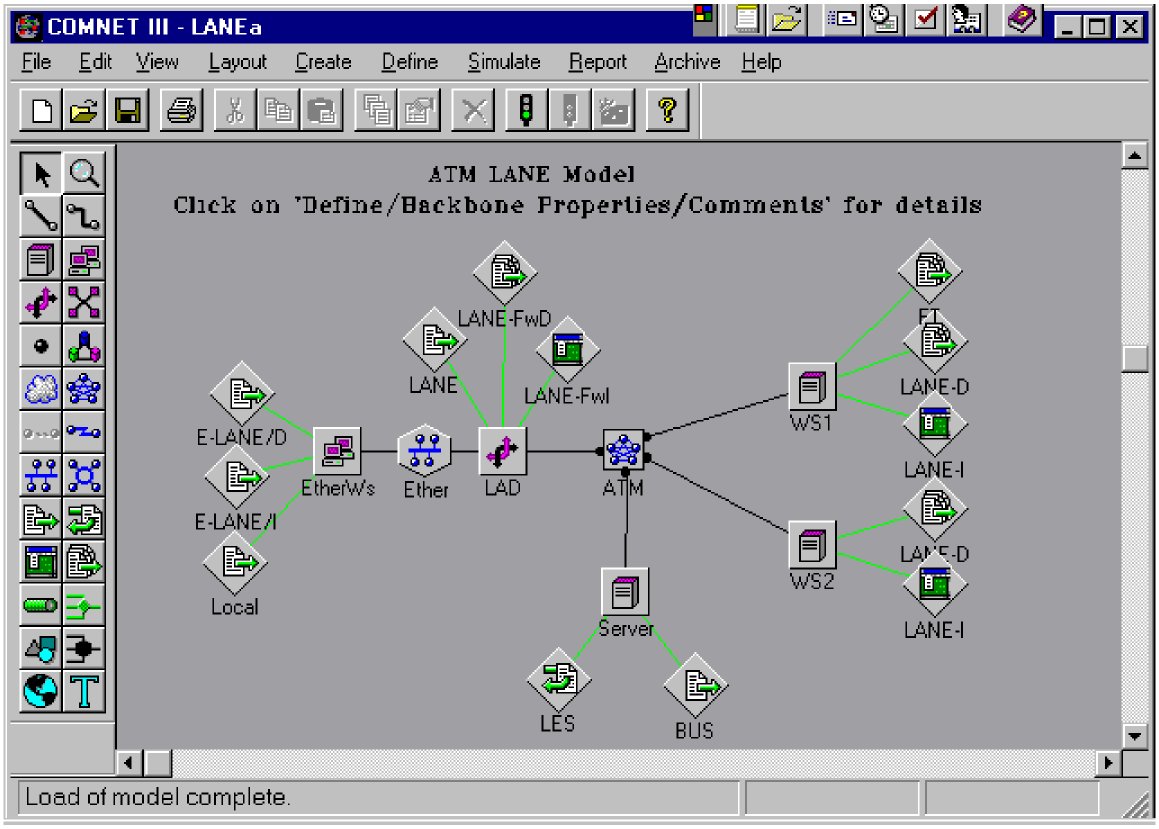
Similarly, the workstations on the traditional LAN do not directly establish a connection with the LECs. Instead, they transmit their messages to the LAD. This is indicated in the model through the two COMNET III message sources labeled ‘E-LANE/D’ and ‘E-LANE/I’ respectively, which are connected to the computer group icon. These sources generate connectionless traffic to the LAD, the size of which is 15000 bytes using the Ethernet protocol. The only difference between both sources is the name of the source, which is subsequently used as a trigger at the LAD.

The source ‘E-LANE/D’ triggers the COMNET III session source named ‘LANE-FwD’, whereas the source ‘E-LANE/I’ triggers the application source named ‘LANE-FwI’. The two forwarding sources are very similar to the sources connected to the LECs. ‘LANE-FwD’ generates 15000 byte-messages with one of the LECs as destination following the Ethernet protocol. Like above, the Ethernet frames are segmented into ATM AAL5 cells on the VLAN. Similarly, the message source ‘E-LANE/I’ triggers the application source ‘LANE-FwI’, which again executes the command sequence

- Send ARP
- Broadcast
- Await Response
- LANE

The parameters for these commands are identical to those described above.

Figure 17 depicts a modified model of such a VLAN. Here, the modules defining the virtual LAN are no longer associated with dedicated physical nodes. Instead, they are represented by the traffic sources only. The VLAN is thus modeled through the relationships between the sources which are established through triggers and the destination lists of the sources. Notice that the model now also contains traditional traffic sources which do not rely on the LANE protocol stack. These are named ‘local’ and ‘FT’ and represent local Ethernet traffic and a file transfer respectively. Also notice that the server now hosts both LES and BUS modules.



ATM LANE Model

Click on 'Define/Backbone Properties/Comments' for details

Load of model complete.

Figure 17: Modified COMNET III LANE model

4.4 Delta Switch Design

The last example illustrates how to use COMNET III to design the switching architecture of an ATM delta switch. Possible design problems include:

- Given the internal architecture of the switch, how large do the buffers have to be sized in order to ensure a certain blocking probability.
- Given the internal architecture of the switch, what traffic levels may be admitted to the switch to ensure a certain QoS for each source

The delta switching architecture consists of a number of simple switching elements which are directly connected to each other. The switching elements are arranged in three layers in this example. Each layer consists of switching elements. The first layer is connected to the input lines. Similarly, the last layer is connected to the output lines. The intermediate layer ensures that all output lines can be reached from all the input lines, and it thus provides an additional layer for switching. This is necessary since each switching element is only connected to two lines on either side. The switching elements can also be interpreted to be very simple single server / single queue systems. Figure 18 shows the internal architecture of the delta switch under consideration in this example.

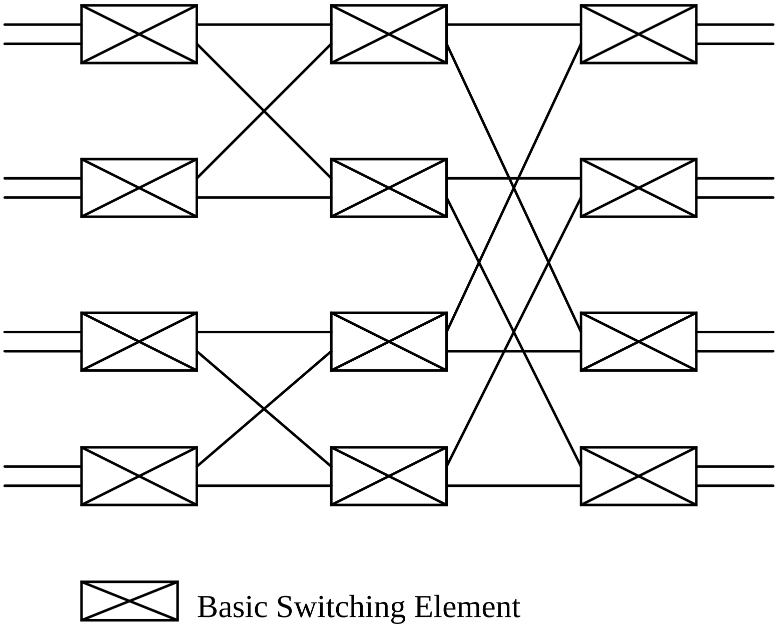


Figure 18: Delta switch architecture

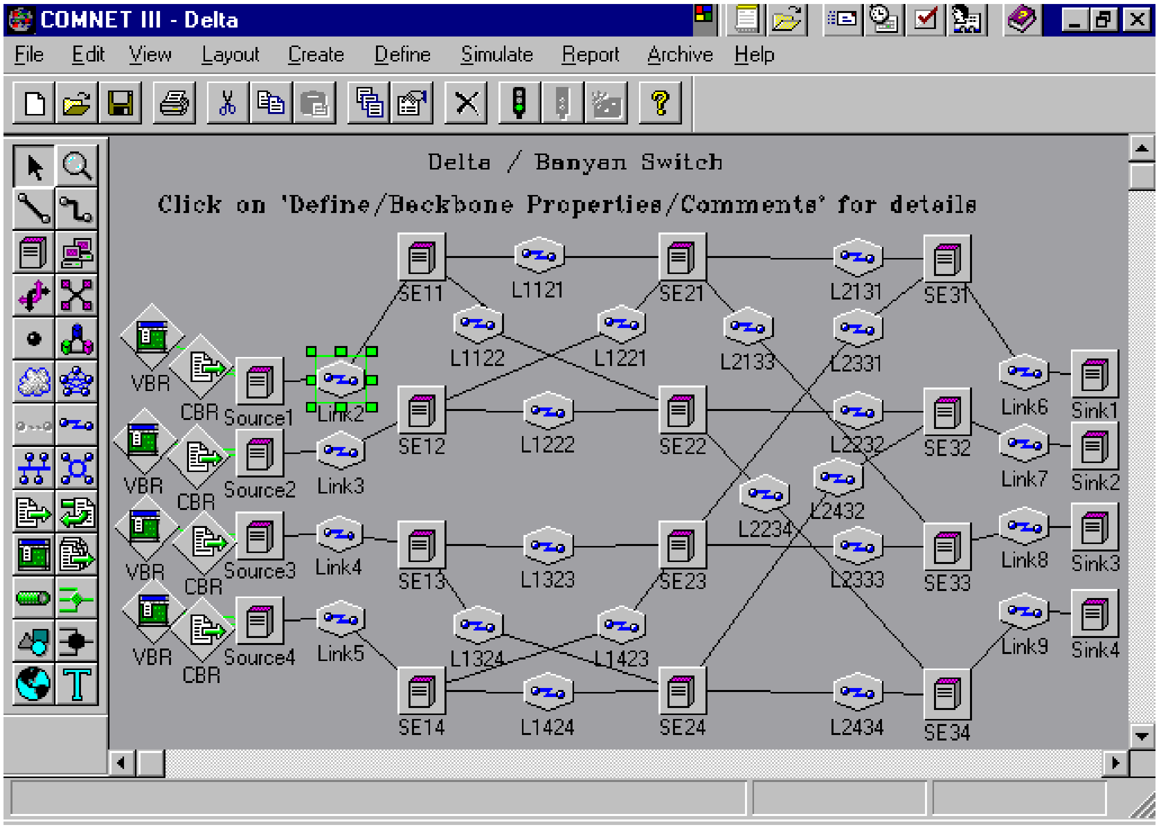
To model this architecture, COMNET III Computer & Communications nodes are used to represent the switching elements. The connectivity between these elements is modeled using the point-to-point link building block. The switching elements are given their default parameter set. This is done in the model to determine the maximum buffer size required to handle different traffic loads. The reports produced by COMNET III will summarize the average and maximum buffer utilization. The processing delay is ignored in this case for simplicity. All other parameters supplied by the Computer & Communications node are not relevant when modeling at this level of detail. Similarly for the point-to-point links. They are given parameter values to indicate the speed of the connections between the switching elements. In this model, the speed has been assumed to be 10Mbps. To ensure that only ATM cells are processed by the switch, the framing characteristics of the point-to-point link have been set to 53 bytes (recall that these are typically used to represent the data link layer PDUs).

Also included in the model are separate Computer & Communications nodes which act a sources and sinks to the traffic. Their only purpose is to represent the external locations where traffic is generated by the sources, or where the traffic terminates. Hence, they too are given the default values. These external sources and sinks are connected to the switch using 155Mbps point-to-point links, representing SONET STS 3c links as used in ATM.

The naming convention of the model is as follows:

- switching elements are labeled ‘SE**’
- connections between switching elements are labeled ‘L****’
- external sources and sinks are labeled ‘Source*’ and ‘Sink*’ respectively
- external input and output lines are labeled ‘Link*’

where * represents a single digit. Figure 19 shows a screen shot of the model.



Delta / Banyan Switch

Click on 'Define/Backbone Properties/Comments' for details

Figure 19: COMNET III model of a delta switch

The traffic defined in this model comes in two forms: a single message source is connected to each source, representing CBR traffic. Similarly, a single application source is connected to each source representing VBR traffic modeled as an ON/OFF process.

Each CBR message source generates 1 cell every 0.000066 seconds, which corresponds to a traffic rate of 6.4Mbps. The inter-arrival time of the source is thus set to 0.000066 seconds at a constant rate. The message size is also determined using a constant probability distribution with value 1. The destination of the source is specified to be one of the sinks on the output side of the switch. The transport protocol has been set to ATM, where the data bytes of the packets has been set to 53 bytes without any overhead. This simplification is made here in the model, since the interest lies in the performance of the internal architecture of the switch, not the end-to-end performance of the cells.

The application source representing the VBR traffic executes the following command sequence:

- OFF
- ON

The OFF command is a global processing command. It represents the time period when the source is in its silent state, i.e. not transmitting any cells. The time delay modeled here is 2500 microseconds, as determined by the parameter value 2.5 milliseconds. After elapsing this delay, the ON command of is executed. This command sends 20 messages / cells with an inter-arrival time between each cell of .000025 seconds, or 250 microseconds. The total delay for the ON period is therefore 500 microseconds. Again, the cells are designated for one of the sinks following the ATM transport protocol where the cell size of 53 bytes is specified. With these values, each VBR source generates traffic at a rate of 2.8Mbps.

This architecture is simulated under different traffic loads by using the COMNET III parameter called ‘Traffic scale’, which can be found under the menu entry ‘Define / Backbone Properties’. Three simulations are run for a total of 5 seconds with the traffic loads of 1, 2 and 2.5, i.e. a total traffic load of 9.2Mbps, 18.4Mbps and 23Mbps per sink. Note that these values have been selected for demonstration purposes only.

The following table summarizes the results between the different traffic loads.

```
                                     Delta
                          NODES: OUTPUT BUFFER TOTALS
                     REPLICATION 1 FROM 0.0 TO 5.0 SECONDS
                             PACKETS         BUFFER USE (P=PKTS, B=BYTES)
NODE                   ACCEPTED  BLOCKED   AVERAGE    STD DEV     MAXIMUM P/B

______________________ ________  _______  __________  _________  _________ ___
SE11                      119735        0         89         82         479 B
SE12                      119736        0         90         83         531 B
SE14                      119736        0         90         83         530 B
SE13                      119735        0         89         82         530 B
SE21                      119238        0         77        115        1007 B
SE22                      120231        0         80        120        1011 B
SE24                      119916        0         79        118        1064 B
SE23                      119553        0         75        112        1166 B
SE31                      119681        0          2         11         116 B
SE32                      120655        0          2         11         116 B
SE34                      119492        0          2         11         116 B
SE33                      119105        0          2         11         116 B
```

```
                                        Delta
                          NODES: OUTPUT BUFFER TOTALS
                     REPLICATION 1 FROM 0.0 TO 5.0 SECONDS
                             PACKETS         BUFFER USE (P=PKTS, B=BYTES)
NODE                       ACCEPTED  BLOCKED   AVERAGE    STD DEV     MAXIMUM P/B
______________________   ________  _______  __________  _________  _________ ___
SE11                         195493        0         394        221        1699 B
SE12                         195493        0         403        227        1702 B
SE14                         195493        0         398        222        1492 B
SE13                         195493        0         396        229        1654 B
SE21                         195831        0         399        398        3291 B
SE22                         195144        0         402        400        3989 B
SE24                         195325        0         387        371        2862 B
SE23                         195651        0         401        408        3561 B
SE31                         196844        0           5         15         116 B
SE32                         195685        0           5         15         116 B
SE34                         194746        0           5         15         116 B
SE33                         194627        0           5         15         116 B
```

```
                                        Delta
                          NODES: OUTPUT BUFFER TOTALS
                     REPLICATION 1 FROM 0.0 TO 5.0 SECONDS
                             PACKETS         BUFFER USE (P=PKTS, B=BYTES)
NODE                       ACCEPTED  BLOCKED   AVERAGE    STD DEV     MAXIMUM P/B
_____________________   ________  _______  __________  _________  _________ ___
SE11                         233350        0        4467       2059       11829 B
SE12                         233351        0        3784       1533       11618 B
SE14                         233351        0        3670       1753       11054 B
SE13                         233329        0        3847       2095       11923 B
SE21                         233514        0        4713       3173       13901 B
SE22                         233049        0        5203       3645       17462 B
SE24                         233568        0        4011       1915       10678 B
SE23                         232971        0        4113       2507       12947 B
SE31                         233514        0           6         17         116 B
SE32                         234197        0           6         17         116 B
SE34                         232233        0           6         16         116 B
SE33                         232844        0           6         17         116 B
```

The above reports illustrate the buffer utilization under the different traffic loads. They clearly indicate that the layer two switching elements require larger buffer sizes. In the initial case, a buffer size of 4000 bytes seems to be sufficient. In the last case, the buffer has to be increased to 18000 bytes to avoid any cell losses under the proposed traffic loads. Notice how the maximum buffer use for the switching elements ‘SE1*’ changes with respect to the switching elements ‘SE2*’. In the first scenario, the use is approximately at half the level of the layer two switching elements. In the last case however, all switching elements of the first two layers show more or less the same utilization. The randomness in the results can be accounted for by the distribution of the destinations. The layer one switching elements seem more constant in their maximum utilization than the layer two switching elements. Also interesting to note is the utilization of the layer three switching elements. Their use never exceeds 116 bytes. This can be accounted for by the fast output speed of the line.

The respective connection utilization are as follows: under the initial scenario, the connections between the switching elements are utilized at approximately 50%. Under the second scenario, this utilization rises to approximately 82%, whereas under the last scenario, the respective value rises even further to 98%. These values provide a partial explanation of the rising buffer utilizations in the switch.

APPENDICES

A. Perspectives on ATM - Brief Review of underlying Concepts

This appendix reviews the main concepts of the ATM protocol and introduces the related terminology. The concepts are introduced by looking at ATM from three different perspectives.

- What is new in ATM?
- The B-ISDN Protocol Reference Model
- ATM Network Topologies and Interfaces

We first of all outline the underlying motivation for the development of ATM, and thus look at the protocol from the perspective of technological innovation. We introduce the principal concepts such as ‘cells’ and ‘service categories’. In the second part of this appendix, we expand on the concept of service categories and introduce the concept of a traffic contract. In the third section, we then change perspective and examine the ATM protocol with reference to the B-ISDN model. This layered model provides an overview of the different functions offered by ATM and outlines their relationships. In section four we adopt the last perspective by concentrating on a typical topology of an ATM network. We also introduce the network interfaces defined for different parts of an ATM network. Finally, in the fifth section we examine alternative architectures for network switches.

A.1 Basic ATM Concepts

The networking technologies of the 80’s can be characterized by two main distinctions: their geographical coverage and their primary traffic characteristics. The first distinction classifies networks according to their geographical coverage into either Local Area Networks (LANs), Metropolitan Area Networks (MANs) or Wide Area Networks (WANs). Each of these network types typically uses specific network protocols and topologies. Bus, ring or star topologies are typical for LANs. Meshed network topologies are used for WANs.

The second distinction is related to Figure 20, which classifies network traffic according to the tolerated transmission delay and the tolerated error rate. On one end of the scale is delay-sensitive traffic such as voice and video. These applications typically generate a steady flow of network traffic which is very sensitive to transmission delays but insensitive to transmission errors. This type of traffic is also known as isochronous traffic. Traditionally, the underlying network technology used for such applications is time-division multiplexing (TDM) based on the traditional digital telephony system. The total bandwidth of a link is divided into small but fixed slots, and the applications can reserve a number of slots per time-frame, and thus be guaranteed bandwidth to fulfill the delay requirements. In case of erroneous transmission, only a small and tolerable amount of information is lost. On the other side of the scale resides the bursty traffic applications, such as file transfers or database request. These are typically characterized by a large volume of data which is generated in bursts, requiring error-free transmission without the strict delay requirements of isochronous traffic. The underlying networking technology in this case is typically based on packet switching and statistical multiplexing. The traffic generated by the application is segmented into variable length packets which are transmitted through the network, incurring uncertain queuing delays and relatively high transmission delays. Extensive error detection and correction functions of the underlying protocols traditionally ensure error-free transmission. Traditional networking technologies are unable to reconcile the two camps, allowing both TDM and packet-switched technologies to co-exists.

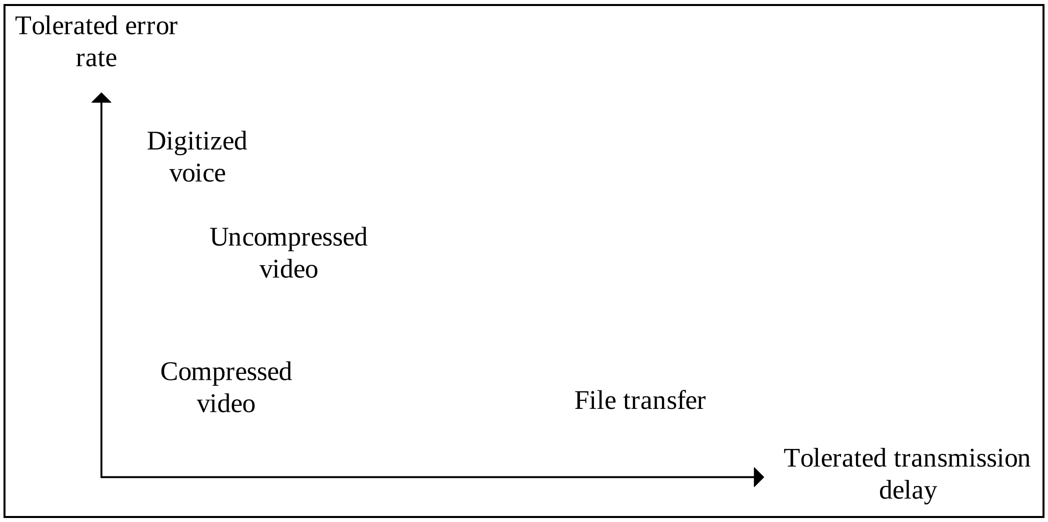


Figure 20: Classification of network traffic based on delay and error requirements

ATM aims to change this worldview. First of all, it attempts to provide high-speed transmission across any geographical distance, hence alleviating the geographical distinction between networks. ATM is designed to be used in what is currently a LAN environment, as well as a WAN environment, with the effect that a distinction based on protocol or topology can no longer be made. Traffic is simply sent out onto ‘the network’, without having to make the distinction between the LAN, MAN or WAN any longer. Secondly, ATM is designed to integrate the different traffic types and has thus been chosen as the underlying protocol for B-ISDN. It will be able to provide the delay-sensitive / error- insensitive services required for isochronous traffic as well as the delay-insensitive / error-sensitive services required for bursty data traffic. Finally, ATM assumes that the physical carriers are based on optical fiber, and hence have an extremely low error rate. It therefore does not make any contingencies for error detection or correction. It is for this reason that ATM is aimed to be the networking technology of the future.

This integration of the different networking technologies is mainly achieved by basing the design of the protocol on the following concepts:

- Asynchronous, connection-oriented transfer
- Fixed-length packets called ‘cells’
- Elimination of error checking functions
- Explicit classification of traffic into service categories

The first three points are essential to providing a high-speed service and therefore to prevent possible performance problems. Connection-oriented transfer implies less overhead when routing the packets. ATM retains the concept of permanent and switched virtual circuits of packet networks, where a logical connection is established between the source and the destination before any data is transmitted. This implies that all data traffic follows the same route, and hence the overhead of finding a route for each packet is eliminated. Using a fixed-length packet, in ATM terminology called a ‘cell’, enables further improvements over the traditional packet-switching protocols. The length of a cell is fixed to 53 bytes, which is short enough to allow the cells to be switched by hardware and hence faster and more economically than through software. Furthermore, the queuing delays are a function of the cell length. Using a short cell size implies a reduction in the overall queuing delay, which in turn enables the transmission of delay-sensitive traffic such as digitized voice. Finally, by assuming a reliable physical transmission channel such as optical fiber, the overhead in existing protocols associated with error checking and flow control can be eliminated, again speeding up the switching process.

The final point on the list provides the functionality to accommodate different service requirements for network traffic. In ATM, the user’s traffic is classified into one of 5 different service categories. Each category provides a different level of service in terms of bandwidth and delay guarantees. The categories are labeled as follows:

- Constant bit rate (CBR)
- Real-time variable bit rate (rt-VBR)
- Non-real time variable bit rate (nrt-VBR)
- Available bit rate (ABR)
- Unspecified bit rate (UBR)

The first category, CBR, is intended for applications requiring absolute bandwidth and delay guarantees. Examples of traffic falling into this category are digitized voice applications or compressed and uncompressed video applications. The two VBR categories have similar requirements in terms of delay and bandwidth, but are typically more variable in nature. Example applications generating VBR traffic include air-line reservation systems, teleconferencing, banking transactions or frame-relay internetworking. The fourth category, ABR, is intended for applications which are less dependent upon the transmission delay and where the user can dynamically control the transmission pattern. They do however require some guarantee for bandwidth and hence transmission. Typical examples here are traditional file transfers, email or critical data transfers. The last category is intended for applications which require no commitments from the network whatsoever, neither in terms of delay nor in terms of bandwidth. File transfer applications could also fall into this category.

The idea behind these service categories is to obtain a better mechanism for managing the limited resources in the network, such as link bandwidth or buffer capacity. The user is assumed to be able to describe the traffic pattern and hence associate it with one of the 5 categories. Knowing the service category, the network consequently has more information about the traffic profile and hence can make a better decision at the time the traffic originates whether sufficient resources are available to provide the required service. This concept of service categories is inherent in almost all the functions of ATM, so let us now take a closer look at some of the details.

A.2 ATM Service Categories and the Traffic Contract

As mentioned above, the idea behind the concept of the different service categories is to exchange as much information as possible between the user - the generator the traffic, and the network - the carrier of traffic, and hence to enable better network utilization. The results of this exchange of information are recorded in what is called in ATM 'the traffic contract'. This concept forms a contractual agreement between the user and the network about the level of service which both have agreed upon. It consists of the following elements:

- A set of traffic descriptors and rules for a compliant connection
- A set of quality of service parameters
- A set of conformance checking rules

These elements are now described in detail.

A.2.1 Traffic Descriptors and Connection Compliance

Traffic descriptors are parameters which capture the inherent characteristics of the user's traffic. They can be interpreted to be an indication by the user to the network about the expected level of transmission that he or she is generating. The ATM standard currently distinguishes between source traffic descriptors and connection traffic descriptors. The source traffic descriptors are those parameters which are negotiated between the user and the network at the time the connection is established. The user typically specifies a desired value and a threshold value, providing a limit to the acceptable values from a user's point of view. When the connection is established, the network attempts to meet the desired value. If this level of service is unattainable due to limited network resources, the network will modify the values indicated by the user in the direction of the threshold value, while guaranteeing that these are not violated. At the end of the negotiation process, the network either refuses to set up the connection if insufficient resources are available, or it will accept to establish the connection with values which lie between the threshold value and the desired value. The following four parameters make up the set of source traffic descriptors:

1. Peak Cell Rate (PCR): the maximum number of cells which the source transmits during a specified time interval.

2. Sustainable Cell Rate (SCR): the maximum average transmission rate for a burst traffic source. This parameter is less than the PCR by definition.

3. Maximum Burst Size (MBS): the maximum number of cells that can be transmitted at once at the peak rate.

4. Minimum Cell Rate (MCR): the minimum number of cells that the source is transmitting during any interval.

The connection traffic descriptors are used by the network while the transmission is in progress. They consist of the above mentioned source traffic descriptors and further specify a Cell Delay Variation Tolerance (CDVT) for both the PCR and the SCR. The purpose of the latter parameters is to place an upper limit on the cell delay variation (jitter), so that the network is given some flexibility to comply with the PCR and SCR values respectively within the bounds of the CDVT. So effectively, they prevent the network from increasing the jitter of a connection to an unacceptable level for the user. These parameters are labeled ‘connection traffic descriptors’ to indicate that they are effective during the transmission phase of the connection.

Not all the service categories require a complete specification of the above parameters. In fact, in some cases the user might not even know the values for some of the parameters. Table 21 summarizes the required traffic descriptors for each of the five different service categories.

	CBR	**rt-VBR**	**nrt-VBR**	**ABR**	**UBR**
PCR	Yes	Yes	Yes	Yes	Yes
SCR	n/a	Yes	Yes	n/a	n/a
MBS	n/a	Yes	Yes	n/a	n/a
MCR	No	No	No	Yes	No
CDVT(PCR)	Yes	Yes	Yes	Yes	Yes
CDVT(SCR)	n/a	Yes	Yes	n/a	n/a

Table 21: required traffic descriptors by service category

The CBR traffic is only described by its PCR and the related CDVT. It places the strictest requirements on the network and implicitly has the highest priority. When a CBR connection is requested by the user, the network will attempt to reserve the bandwidth and buffer requirements specified by the PCR. The user does not need to specify any further parameters. In case of the VBR service class, the implied priority is lower than for CBR, so the user is required to also specify the SCR and its related CDVT. Given the VBR service category and values for the required parameters allows the network to make guarantees if the resources are available. The ABR and UBR service categories implicitly have the lowest priorities, with ABR priorities exceeding UBR priorities. From the network's point of view the two categories are only allocated to those resource which have not been committed to the CBR and VBR classes. For this reason they are sometimes referred to as 'best effort' service classes. There is still a considerable difference between the two classes.

First of all, contrary to the generalization made in section A.1, any traffic categorized as UBR is sent using a connectionless protocol. Thus, for UBR traffic no logical connection is set up between the source and the destination. For this reason only the PCR and the desired variation are required to describe the traffic. Since it is given lowest priority with no guarantees, those two parameters are sufficient for the network to manage its resources.

Secondly, the ABR service category is considered as a special case in the specification. Like UBR it is intended to use up any available bandwidth / buffer resources which have not been allocated already. Like CBR and VBR, it is still transmitted using connection-oriented transfer, thus allowing the user to obtain some transmission guarantees from the network. However, in return for these guarantees, the user is assumed to be able to control the pace of the transmission. For this reason the ABR service category specifies flow control mechanisms where the network directs the source to reduce its transmission in times of heavy congestion. In addition to the traffic descriptors listed in Table 21, additional parameters are thus required to specify the flow control mechanisms. These parameters are described in detail below under the section B.2.3.

A.2.2 Quality of Service Parameters

The second component of the traffic contract is related to the quality of service (QoS). The QoS parameters provide an indication to the user about the level of service that he or she can expect to receive from the network. These parameters describe the end-to-end performance of the connection. Like the traffic descriptors, the actual values of some of these parameters are negotiated during the connection establishment phase of the call. The user specifies the desired QoS and the network either agrees to the service level, re-negotiates the values or declines the connection.

Version 4.0 of the ATM Forum's traffic management specification currently defines the following QoS parameters:

1. Maximum Cell Transfer Delay (Max CTD): the cell transfer delay is defined to be the transmission delay of a cell between the source and the destination. Included in this measurement are the link delays along all the links on the end-to-end path, as well as all the ATM node processing delays. This parameter puts an upper bound on the end-to-end transfer delay for all cells.

2. Mean Cell Transfer Delay (Mean CTD): this parameter indicates the average end-to-end transfer delay of cells between the source and the destination.

3. Cell Delay Variation (CDV): this measures the variability of the cell transfer delays, also known as jitter. The difference between the CDV and the CDVT is as follows: the former measures the actual jitter, whereas the latter is used in the context of the GCRA and therefore has an impact on whether the cell is conforming or not. The next appendix will clarify some details on the GCRA.

4. Cell Loss Ratio (CLR): this parameter is defined by the ratio of lost cells to the sum of all the successfully transmitted cells (i.e. this excluded all dripped and errored cells).

5. Cell Error Ratio (CER): this parameter is defined as the ratio of errored cells to the sum of the transmitted cells (i.e. it includes successfully and errored cells, but excludes all dropped cells).

6. Severely Errored Cell Block Ratio (SECBR): this parameter measures the number of cell blocks in which cells have been lost, misinserted, or errored as a ratio of the total number of transmitted cells. A cell block is simply defined to be a sequence of user cells (typical block sizes are 28, 256, 512 or 1024 cells). If the number of errored, misinserted or lost cells within that block exceeds a certain threshold, then the cell block is counted as severely errored.

7. Cell Misinsertion Rate (CMR): this parameter counts the number of misinserted cells over a specified period of time and expresses it as a ratio. A cell is typically 'misinserted', if its header contains an error which is not detected and leads it subsequently to be transmitted along a different connection.

Only the Max CTD, the Mean CTD, the CDV and the CLR are part of the negotiation process which takes place when the connection is established. The latter three parameters in the above list are simply calculated to indicate the QoS that the call has received from the network. The ATM Forum actually specifies 4 different classes for the QoS. Class 1 is intended for circuit-emulation, class 2 for variable bit rate audio and video traffic, class 3 for connection-oriented data transfer and class 4 for connectionless data transfer. These QoS classes map onto the service categories described above as shown in Table 22.

	CBR	**rt-VBR**	**nrt-VBR**	**ABR**	**UBR**
QoS Class	1	2	3	unspecified	4
Max CTD	Yes	Yes	No	No	No
Mean CTD	No	No	Yes	No	No
CDV	Yes	Yes	No	No	No
CLR	Yes	Yes	Yes	Yes	No

Table 22: required QoS by service category

This table further illustrates the distinction between the different service categories. Both CBR and VBR provide a high level QoS and therefore make use of all parameters except for the mean CTD. The lower priority categories on the other hand provide little or even no support for the quality of service.

Once these parameters have been negotiated during the connection establishment phase, the network commits to maintaining this QoS, provided the source itself does not violate the values of the traffic descriptors. In other words, whenever a new call wished to transmit, the network must ensure that the QoS of all existing calls are not violated by the admittance of the new call.

A.2.3 Conformance Checking Rules

The third component of the traffic contract specifies the rules by which the network is going to check the conformance of a user cell. One of the purposes of the traffic descriptors and the QoS parameters is to negotiate the service level between the network and the user in the traffic contract. Like any contractual agreement, penalty clauses are specified to cover the case of non-compliance. In this case, non-compliance means that a source agreed to transmit at a certain rate, for example the PCR, but in reality exceeds this transmission rate at an unacceptable level for the network. For this reason, the source and the network agree on the definition of a compliant connection. This includes a precise definition of a conforming cell, a dropped cell, a delayed cell etc. This definition is typically based on the generic cell rate algorithm (GCRA) with which the network supervises the transmission rate of the source. Again, the definition differs between the service categories. Details on the algorithm as well as the cell dropping procedures for different service classes are given below in appendix B.

A.3 The B-ISDN Protocol Reference Model

The previous sections looked at ATM from the point of view of the technical innovations and provided a detailed introduction to some of the new concepts. In this section, we will look at the protocol from a formal point of view by examining the B-ISDN reference model. This model is depicted in Figure 21.

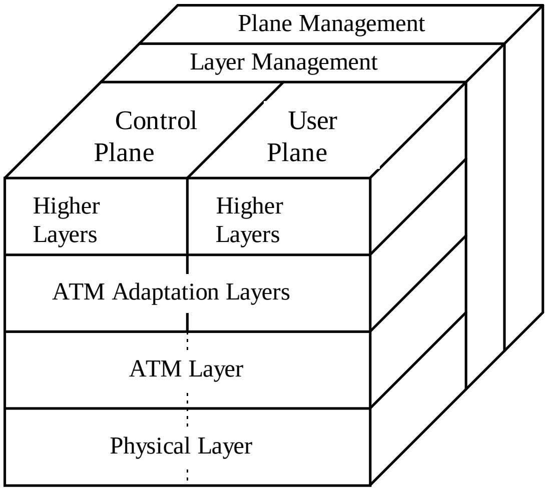


Figure 21: the B-ISDN reference model

The B-ISDN model has two dimensions: a layered dimension illustrating the relationships of the different protocol layers, and a plane dimension illustrating the three prevalent traffic planes. Let us start by looking at the plane dimension.

The model consists of three different planes: the user plane, the control plane and the management plane. These three planes represent the different types of traffic which typically flow through a network. User traffic is represented by the user plane, signaling traffic is represented by the control plane and management traffic is represented by the management plane.

The user plane is responsible for the transmission of the user's data. This typically includes the transmission of the user's data, as well as performing flow control and error recovery operations. The control plane is responsible for setting up a connection between the source and the destination, also known as connection admission control (CAC). It has to determine whether a newly arriving call can be admitted to the network and it negotiates the parameters specified in the traffic contract. Subsequently, the control plane has to maintain this connection, which is called usage parameter control / network parameter control (UPC/NPC). This function ensures that the source and the network comply with the traffic contract and it initiates corrective actions if the traffic contract is violated. The control plane is also responsible for releasing a connection. The management plane is actually divided into layer management and plane management.

The former is responsible for the provision of operations, administration and maintenance (OAM) functions. These include categories such as:

- fault management
- performance management
- security management
- accounting management
- configuration management

Basically, the layer management functions are responsible for the collection of management statistics to be stored in management information bases (MIB's). The plane management functions are responsible for the coordination of all planes.

The layers in the B-ISDN model illustrate the protocol stack adopted by ATM. As indicated in Figure 21, the ATM protocol is generally considered a layer 2 protocol within the OSI reference model. It is located on top of the physical layer and requires the services of higher layer protocols. However, strictly speaking there is no direct correspondence between the two models, since ATM also provides some services which are provided by layer 3 protocols in the OSI model.

Since the functionality implemented by the different layers provide a good understanding about the operation of ATM, we will now look at the lower three layers of the B-ISDN reference model in detail.

A.3.1 The ATM Adaptation Layer

The higher layers in Figure 21 represent all the protocols which are located on top of ATM. These typically include protocols such as TCP/IP, IPX or ftp in the user plane, SNMP or CMIP in the management plane or Q.2931 in the control plane. Before the higher layer protocol data units (PDU's) are handed to the ATM layer, they are required to pass through the ATM Adaptation Layer (AAL). This layer adapts the transfer process of the upper layer services to the ATM layer services. In particular, the AAL is responsible for the segmentation and reassembly (SAR) of higher layer PDU's and the execution of convergence functions(CS). It takes the higher layer PDU and segments it into a 48 byte AAL-PDU and provides some error checking functions. The ATM Forum currently distinguishes between four different types of AAL's to cater for the five ATM service categories. Table 23 outlines this relationship and summarizes the main characteristics of the different AAL's.

Class	A	B	C	D
AAL Type	AAL1	AAL2	AAL3/4, AAL5	AAL3/4, AAL5
Possible Corresponding Service Category	CBR	rt-VBR	nrt-VBR, ABR	UBR
Bit Rate	Constant	Variable		
Timing relation between source and destination	Required		Not Required	
Connection Mode	Connection-oriented			Connectionless
Service Examples	DS1 / E1 emulation, CBR speech / video	packet video / audio	FR, X.25	TCP/IP, SMDS

Table 23: AAL Classes and their Service Characteristics

AAL1 performs the functions which are required to adapt CBR services to the ATM layer services. On the transmitting side, the convergence sublayer of AAL1 takes the PDU's from the higher layers and adjusts the cell delay to ensure a constant bit rate traffic stream. The SAR function of AAL1 takes 47 bytes of payload from the higher-layer PDU and adds 1 byte of overhead before handing the resulting 48 bytes to the ATM layer. As shown in Figure 22, the overhead bytes contains fields for sequencing and error checking. On the receiving side, the SAR function re-assembles the higher-layer PDU by stripping the 1-byte header off. The CS sublayer then provides the following services, using the information carried in the AAL1 header:

- detecting lost or missequenced cells
- adjusting the cell delay to generate a constant bit rate
- providing error correction on the AAL1 header
- providing error correction for the user-payload of high-quality video/audio applications
- re-synchronizing the clock frequency at the receiver if this is necessary

Sequence number (1/2 byte)	Sequence number protection (1/2 byte)	Payload (47 bytes)

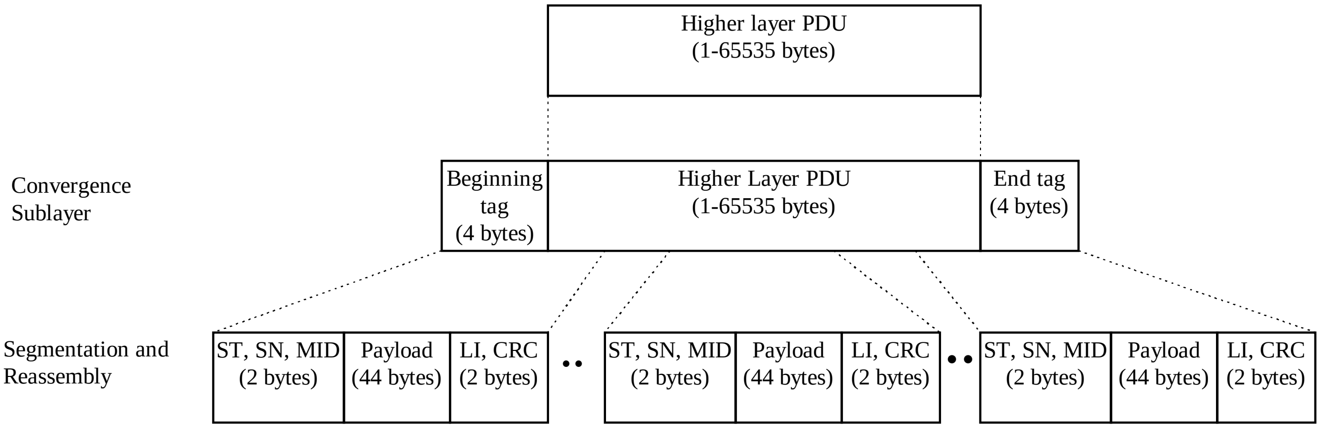
Figure 22: AAL1 PDU Format

AAL2 is intended for traffic which requires a timing relationship between the source and the destination, such as MPEG video streams. Unfortunately, the ATM Forum has not yet completed the specification of the AAL2 services, and so many details are still undetermined. Figure 23 provides an indication of the AAL2 PDU. However, the length of the different fields as well as their exact functionality are not clear to date. As illustrated, the AAL2 PDU consist of both a header and a trailer. The header contains a sequence number, which is used as above, and an information type field. This field is useful for indicating whether the payload carries the beginning, continuation or the end of a message. The trailer consist of a length indicator and a CRC. The length indication makes provisions for variable length user data carried in the payload. The CRC provides error detection and recovery functions.

Sequence number	Information type	Payload	Length indicator	CRC

Figure 23: AAL2 PDU Format

The third adaptation layer type is actually labeled AAL3/4. It provides transport services for variable length frames, including error detection functions. The ATM Forum initially considered two separate AAL types for variable bit rate traffic: one for connection-oriented VBR traffic and one for connectionless VBR traffic. Due to their resulting similarities, these two types have been merged into one - hence the label AAL3/4. Both connectionless and connection-oriented services are allowed. The connection-oriented service is inherent in the ATM protocol. The connectionless service is provided here using a connectionless-server, giving the flexibility to carry bursty LAN traffic or any data which is tolerant with respect to delay variations. The AAL3/4 functionality is depicted in Figure 24. The convergence sublayer adds a header and a trailer of 4 bytes each to the higher layer PDU. These contain fields to tag the beginning and end of the message to provide additional error checking. The AAL3/4 then segments the higher layer PDU into 48 byte AAL3/4 PDU’s, adding a header and trailer to each PDU. As shown in Figure 24, the format here is similar to the proposed AAL2 format, containing fields for the segment type, the sequence number, the length indicator and the cyclic redundancy check (CRC). In addition, it specifies a message identifier field, allowing the transmission of multiple simultaneous packets on a single connection.



Key:

ST	Segment Type (BOM, COM, EOM)	LI	Length Indicator
SN	Sequence Number	CRC	Cyclic Redundancy Check
MID	Message Identifier		

Figure 24: AAL3/4 PDU Format

The last type, AAL5, is also intended to service variable bit rate traffic. However, it differs from AAL3/4 in that it provides less functionality and therefore a higher efficiency. Since this type supports connectionless services, there is no need for providing sequencing functions in this layer. Furthermore, instead of using a separate field to indicate the beginning, continuation or end of the user's message, a flag in the ATM layer header is used. With these modifications, the AAL3/4 header and trailer can be eliminated, making room for additional payload in the ATM cell. This is indicated in Figure 25. Notice that the convergence sublayer of AAL5 adds a trailer to the higher layer PDU. Like above, this trailer provides additional error checking functions.

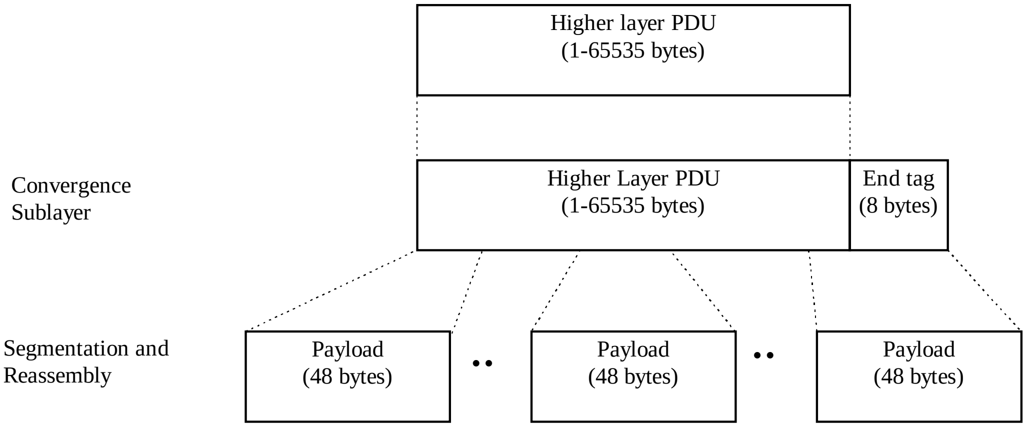


Figure 25: AAL5 PDU Format

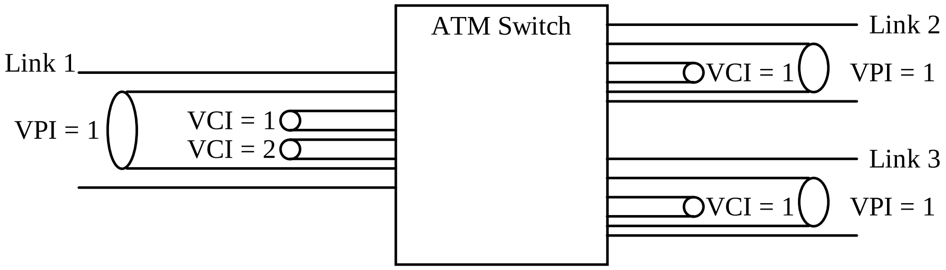
A.3.2 The ATM Layer

The purpose of the ATM layer is to enable the transmission of cells across the network. At this layer, a 5 byte header is added to the AAL PDU, as illustrated in Figure 26. The fields in this header enable three basic functions: switching cells, flow control and error control.

Generic Flow Control (1/2 byte)	Virtual Path Identifier (1 byte)	Virtual Channel Identifier (2 bytes)	PTI, CLP (1/2 byte)	Header Error Control (1 byte)	Payload (48 bytes)

Figure 26: ATM Cell Format

The switching function in ATM is based on the concepts of a Virtual Path (VP) and a Virtual Channel (VC). A virtual channel is used to carry customer data, each having their own service levels. The virtual path is used to bundle a number of virtual channels with the same source and destination together. In this way, related VC can be explicitly connected and possibly switched more efficiently. For example, a video-conference between two sites might require three different VC's to carry the video, voice and data traffic. The different traffic types might be transmitted over CBR, VBR and ABR VC's respectively. All three VC's can be bundled into a single VP between the source and the destination. The virtual channels and virtual paths are established using the Virtual Path Identifier (VPI) and Virtual Channel Identifier (VCI) fields in the ATM cell header. Each switch along the end-to-end path maintains a VPI/VCI switching table. This table contains relates the VPI/VCI values on incoming links to the VPI/VCI values on the outgoing links and thereby provides the switching functionality of the ATM layer. The values in the table are established at connection-setup time. For a PVC, they are typically set by the network manager, for an SVC, they are determined by the CAC function (see below for details). Whenever a cell arrives, the VPI/VCI values in its header are examined and the cell is switched to the relevant output link according to the values in the switching table. The cell is transmitted using the VPI/VCI values of the outgoing link until it reaches its destination. This process is illustrated in Figure 27.



Link in	VPI	VCI	Link out	VPI	VCI
1	1	1	2	1	1
1	1	2	3	1	1

Figure 27: VPI/VCI Switching

The generic flow control field in the ATM cell header is used to support flow control and multiplexing functions. The ATM Forum has not yet completed the standardization process for this field. Currently, it is typically used by the GCRA to provide congestion information.

The payload type identifier (PTI) gives an indication about the type of data carried in the payload. ATM cells may be used to carry user data, signaling data or management data, each of which take a different value in the PTI field.

The cell loss priority (CLP) field in the header provides important information about the priority level of the cell. The user may mark this field as either 0 or 1, the former indicating a higher priority than the latter. In case of heavy network congestion, the network may decide either to immediately drop lower priority cells, or change the CLP bit of certain data streams to 1 such that they are more likely to be dropped subsequently. The details of this procedure will be outlined below under the section on UPC/NPC.

Finally, the header error control (HEC) provides a checksum for the header, allowing for the detection of errors and even the correction of 1-bit errors. Note that this only applies to the cell header.

A.3.3 The Physical Layer

The physical layer is responsible for transmitting the ATM cells over a physical medium across the network. It’s functions are separated into the Physical Media Dependent Sublayer (PMD) and the Transmission Convergence (TC) sublayer. The PMD provides the actual transmission functions, whereas the TC ensures that a steady flow of bits and bytes is generated for transmission.

ATM currently supports many different physical medium types, some of which are listed below in Table 24. Note that the effective bit rate is typically lower than the actual bit rate, due to in-band or out-of-band signaling and frame overheads at the physical layer.

Name	**Bit Rate (in Mbps)**	**Effective Bit Rate (in Mbps)**
SONET STS-1	51.84	49.536
SONET STS-3c	155.52	149.76
SONET STS-12c	622.08	594.432
SDH STM1	155.52	
SDH STM4	622.08	
DS-1	1.544	1.536
DS-2	6.312	6.176
DS-3	44.736	40.704
E1	2.048	1.92
E3	34.368	
E4	139.264	
Multimode Fiber(FDDI)	100	
STP	155.52	
Fiber Channel	155.52	
UTP	51.84 25.92 12.96	

Table 24: Physical Layer Interfaces

A.4 ATM Network Interfaces

So far, we have only introduced the main concepts and the ATM protocol layers without looking at a typical topology. In this section we start to take a global view of an ATM network. As shown in Figure 28 and implied in section A.3.3, an ATM topology is typically based on point-to-point links. No provisions are made in the protocol specification for medium access control or multi-access links. Also shown in this figure are the different types of nodes within an ATM network. ATM routers or workstations connect directly to ATM switches. The routers provide the connectivity between existing network protocols to the ATM network. In particular, they are responsible for mapping the network addresses from other protocols to those used by ATM. The switches make up the ATM backbone network. Some of the existing switch architectures are discussed in the next section.

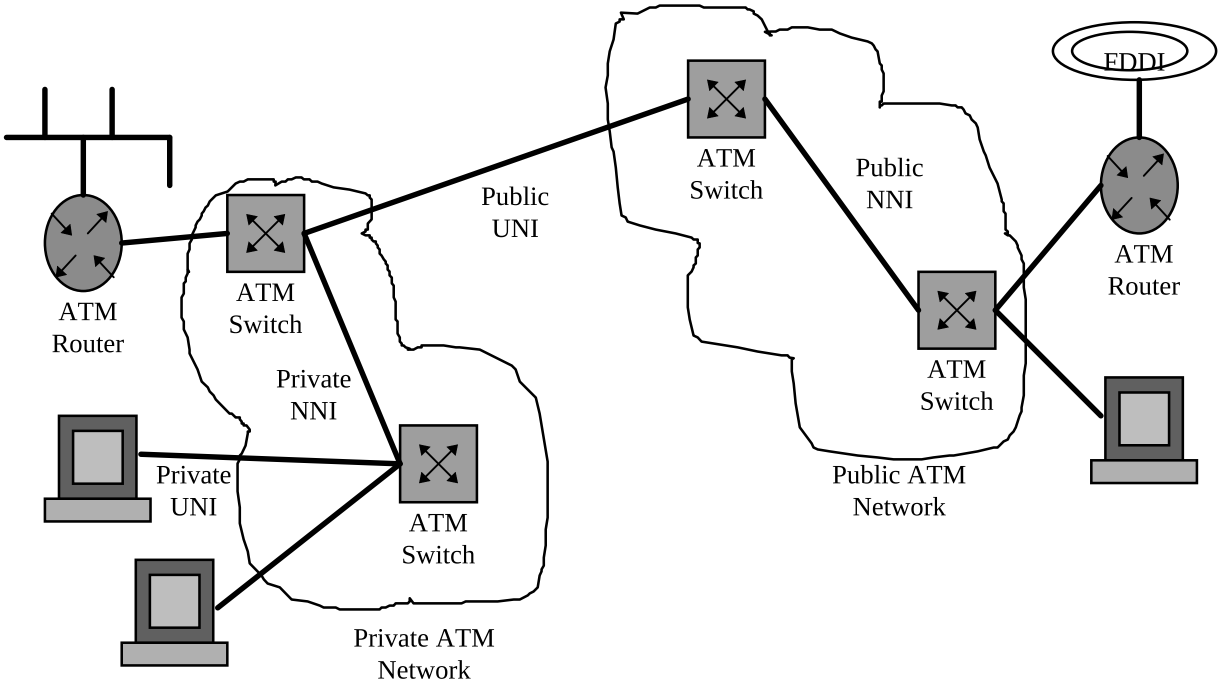


Figure 28: Typical ATM topology showing the network interfaces

Figure 28 also highlights four different network interfaces:

1. Private user-network (UNI) interface
2. Public UNI
3. Private network-node (NNI) interface
4. Public NNI

A private UNI defines the interface between the user's equipment and a private ATM switch. The public UNI defines the interface between a private ATM switch and a public ATM switch. Similarly, a private NNI defines the interface between two ATM switches within a private ATM network. A public NNI defines the interface between two ATM switches within a public ATM network. These distinctions not only permit a more precise discussion about the different elements of an ATM network. They also emphasize the different protocols and cell formats which are used within an ATM network. For example, the management functions required for the public UNI differ from those required for the private UNI. Similarly, signaling functions across the UNI differ from those at the NNI. The distinction between the interfaces is outlined here because some of the explanations below are specific to particular interfaces. For example, the CAC and UPC/NPC functions discussed in the following appendix relate particularly to the UNI. However, to pinpoint the precise differences between these interfaces is beyond the scope of this brochure.

The relationship between the B-ISDN model and the network topology is illustrated in Figure 29. The functions described in the previous section are located within individual ATM nodes. Typically, the connections across the private UNI implement the functionality of all the layers of the B-ISDN model. The connections between nodes across the NNI or the public UNI only implement the ATM layer and the physical layer functions.

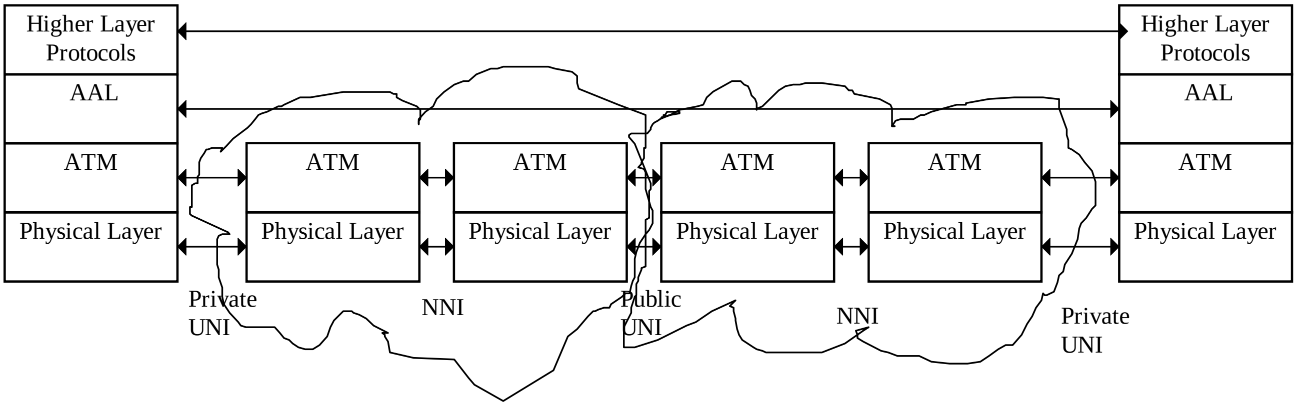


Figure 29: Interactions between ATM protocol functions across network interfaces

A.5 ATM Switching Architectures

The last section in this appendix briefly outlines some of the existing internal switch architectures. These make a significant contribution to the overall end-to-end transmission delay and so the internal switch architecture has to enable and support the high-speed transmission services of ATM networks. A switch typically consists of a number of input buffers, a switching module and a number of output buffers. The input and output buffers are responsible for storing the cells, either on the input or the output side. The switching module is responsible for switching the cells from the input buffer to the output buffer. The four main design approaches for ATM switches are

- Crossbar switches
- Delta switches
- Shared memory switches
- Shared bus switches

Crossbar switches consist of a matrix of lines connecting the input ports to the output ports. The lines connect to each other at crosspoints to provide the connectivity between all the input ports to all the output ports, as shown in Figure 30. This implies that the size of crossbar switches is limited, since the number of crosspoints increases quadratically with the number of input and output ports.

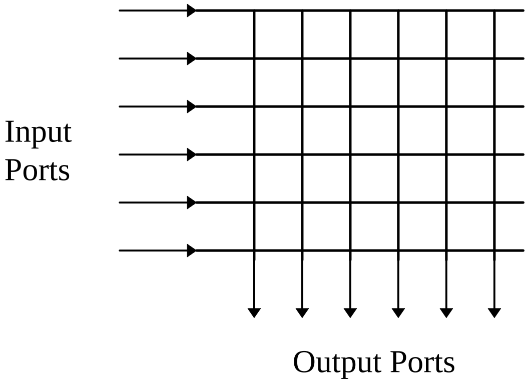


Figure 30: Crossbar switch

Delta switches consist of a series of interconnected switching units. These switching units are typically arranged in layers. The first layer provides the connectivity to the input ports and switches the incoming cells through to the second layer switching units. The intermediate layers perform similar switching functions. The last layer of switching units provides the connectivity to the output ports. Each individual switching unit may have buffering capabilities. Depending on the connectivity between the different layers of switching units, delta switches are either classified as blocking or non-blocking switches. If the aggregate bit-rate of the outgoing connections on a switching unit exceeds the aggregate bit-rate of the incoming connections on a switching unit, the switch has non-blocking capabilities. Otherwise, cells may be blocked inside the switch, and the individual switching units typically have buffering capabilities. Figure 31 illustrates the basic design of a delta switch.

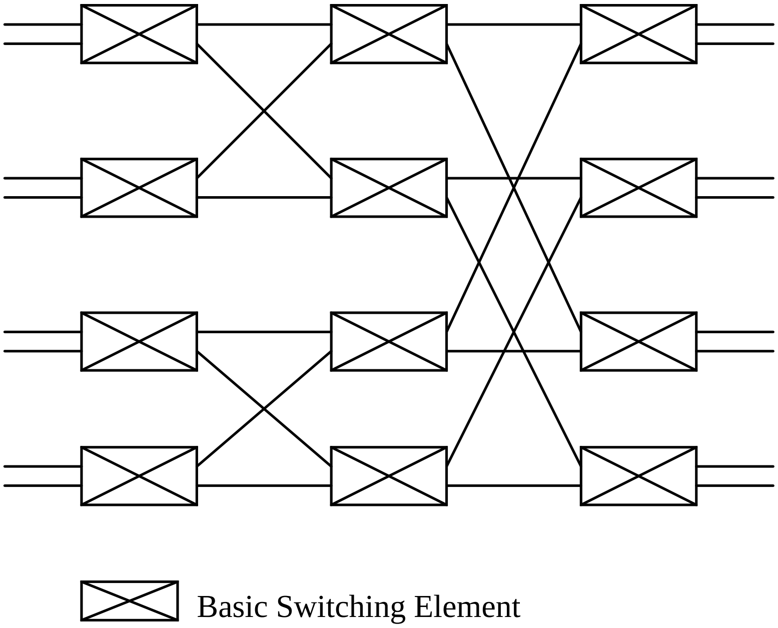


Figure 31: Delta switch architecture

Shared memory switches are characterized by a central memory in which the cells are buffered. Typically, the design includes a multiplexor on the input side and a demultiplexor on the output side. Incoming cells are multiplexed and sent through a single connection to the shared memory. They are typically buffered by VPI/VCI value. On the output side of the switch, the demultiplexor takes the cells from the memory and distributes them to the respective output ports. The basic architecture of a shared memory switch is shown in Figure 32.

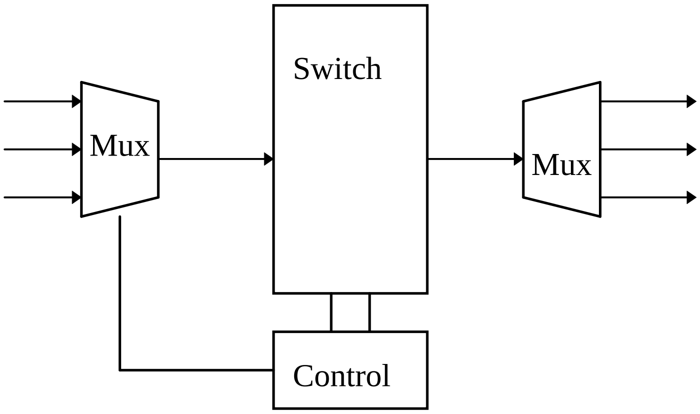


Figure 32: Shared memory switch

Shared bus switches use an internal bus to connect the ports. They are further categorized into single bus or multiple bus architectures. As the name implies, a single bus architecture contains only one bus through which the ports are connected. A multiple bus architecture provides some level of parallelism by connecting the ports through multiple busses. The speed of the busses is typically in the gigabyte range. In case of a single bus architecture, a medium access algorithm arbitrates the access of the ports to the bus. For multiple bus architectures, a separate bus is provided for each port, hence eliminating the need for an arbitration algorithm. Figure 33 illustrates a single bus architecture.

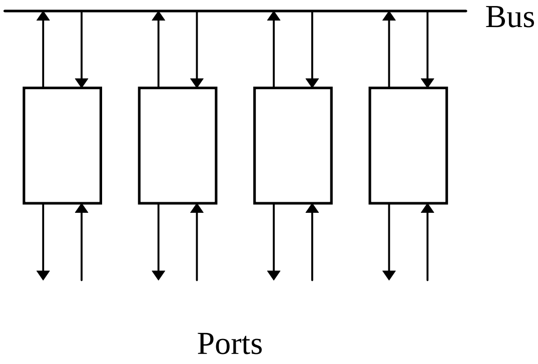


Figure 33: Single bus architecture

In addition to the internal switching architectures outlined above, current ATM switches also differ in their buffer and queuing characteristics. Regarding the buffering options, cells may be buffered either at the input side, internally or at the output side of the switch. Input buffering may lead to the problem of head-of-line blocking. If the output link of the first cell in the input buffer gets blocked, it has to remain in the input buffer and hence prevents all following cells from being switched, even if their respective output ports are unblocked. Other switches, for example delta switches, implement internal buffering. Such designs enable scalability, but typically restrict the network managers from configuring other functions such as priority queuing or multicasting. Output buffering seems to overcome both those problems, providing the network managers with configurable options as well as avoiding head-of-line blocking. Most switches therefore resort to output buffering or even a combination of input, internal and output buffering.

Regarding the priority options, current ATM switches differ by their support for configurable queuing and priority functions. Some switches allow the network manager to configure the queuing and priority options. Others provide hard-wired priority and queuing configurations. The shared memory architecture, for example, may allow a queuing strategy based on the service class, where different queues are provided for CBR, rt-VBR, nrt-VBR, ABR and UBR traffic. The queues might be serviced according to their priority, thereby controlling the throughput characteristics of the switch.

B. ATM Services in Detail

The previous appendix provided an overview of the ATM protocol and explained the principal concepts. In this appendix, we expand on those issues which require further explanations. In particular, we discuss the details of the control plane of the B-ISDN model by taking a closer look at the CAC functions and UPC/NPC. We then describe the operations of fault and performance management, both of which are services of the management plane within the B-ISDN reference model.

Note that the purpose of this appendix is to expand on those details of the ATM protocol which are important from a modeling perspective. These areas have to be particularly well understood in order to be modeled. This provides the principal motivation for this appendix. The following appendix on modeling ATM concepts is based upon the details described in this appendix and makes considerable references.

B.1 Connection Admission Control (CAC) and Resource Management

Connection Admission Control (CAC) is the traffic management function performed by the network to determine whether a newly arriving call should be admitted to the network or not. One of the main considerations here is that the QoS for existing connections are under no circumstances jeopardized. Thus, at either establishment time of a SVC or a PVC, the network has to consider the availability of network resources along the entire path from the source to the destination. The limiting network resources under consideration here are trunk bandwidths, buffers at the switches as well as internal switching resources required to manage the overhead of each additional call (e.g. managing VPIs / VCIs). A further responsibility of CAC is to determine the values of some of the parameters in the traffic contract, which are subsequently required for UPC/NPC, such as the CDVT.

Before we discuss the details of these functions, two considerations should be made concerning the connection type and the service classes. First of all, the CAC functions are mainly used for setting up SVCs. PVCs are typically established manually by the network manager and therefore require less support from the network protocol. This still implies that any connection established in an ATM network has to respect the QoS of existing connections. Secondly, a difference is made between the ATM service classes. The ATM Forum currently does not specify CAC functions for UBR traffic. Furthermore, the specification for the remaining classes leaves considerable room for interpretation and therefore vendor differentiation. For this reason some of the explanations made below seem generic and in some cases, the main alternatives are outlined.

CAC can be separated into two principal functions: managing the network’s resources by protecting the QoS of existing calls and finding a route for newly arriving calls from the source to the destination. These are explained in the following two sections.

B.1.1 Resource Management Policies

The first resource management policy discussed here can be labeled ‘overbooking’. Depending of the mix of traffic services, the network may decide to overbook some of the resources. The idea is similar to an airline reservation system. The majority of service classes are intended for variable bit rate traffic, where the transmission cell rate is bursty. The efficiency of the switch can be increased by statistically multiplexing several connections over the same trunk, thereby exploiting the variability of the arrival pattern of the bursty traffic. This is particularly useful if the sources are able to provide an indication of the variability of the traffic (hence the motivation for the set of traffic descriptors for the VBR and ABR service classes).

The problem of trunk bandwidth management can be illustrated by considering different buffer sizes at the switches. The easiest reservation mechanism would be to use the PCR as an indicator, reserving as much bandwidth as specified by the PCR indicator. This is indeed the case for CBR applications. For VBR applications however, this leads to an inefficient use of the available bandwidth, since for these applications the difference between the PCR and the SCR is typically considerable. The trunk makes reservations based upon the PCR, the VBR applications however, only make use of a fraction of this reserved bandwidth as indicated by the SCR. The bigger the difference between PCR and SCR, the more bandwidth is inefficiently used. Overbooking is one way to alleviate this inefficiency. However, this only works if the variability of the cell rate is small. With a high cell rate variability, the probability increases that many statistically multiplexed sources transmit at their peak burst size at the same time, leading to congestion at the switch.

A similar management problem occurs for switch buffers. If the switch only has small buffers (say 100 cells), it is more likely to drop cells. The network therefore has to reject more calls in order to maintain the QoS of the existing calls, in particular the cell losses. Rejecting calls however implies a decrease in the utilization rate of the links. If the switch has large buffers (1Mb), the probability of dropping cells at high traffic levels is reduced, but additional buffering delays are introduced. This in turn is unacceptable for any real-time traffic which is subject to tight delay constraints. This discussion illustrates the trade-off between buffer size and link bandwidth utilization: a larger buffer has the advantage of better utilizing the link bandwidth while maintaining the CLP. However, the disadvantages are that the buffering delay is increased and the overall buffer utilization declines.

An alternative algorithm for resource management is Fast Resource Management (FRM). It consists of fast bandwidth reservation and fast buffer reservation. Upon arrival of a call, a FRM cell is transmitted from the source to the destination, containing the resource requirements of the respective call. Each switch along the end-to-end path can then reserve the required resources or reject the call.

Fast bandwidth reservation can operate in two modes: immediate unguaranteed and delayed guaranteed. The former starts the transmission of user traffic with the FRM cell containing the resource requirements, immediately followed by the user traffic. Each switch inspects the FRM cell and decides whether to accept the call or not. Since the traffic burst follows immediately after the cell, no guarantee can be made about the acceptance of the call. The latter mode, delayed guaranteed, transmits only the FRM cell to the destination. Each switch along the end-to-end path once again makes an accept/reject decision, either reserving the required bandwidth for the call or dropping the FRM cell. The user traffic is only transmitted after the bandwidth has been successfully reserved. Upon completion of the call, the resources have to be released again using a FRM cell. Loss or corruption of the release cell by the network would be equivalent to lost bandwidth. Another problem with delayed guaranteed FRM arises with long transmission delays. As soon as the FRM cell has been accepted by a switch, the reserved bandwidth becomes unavailable to other calls. If the remaining reservation process is delayed due to large round-trip delays, this bandwidth remains unused, hence introducing inefficiency in the network.

The same principles can be applied for fast buffer reservation. Figure 34 outlines the mechanism by example of fast bandwidth reservation.

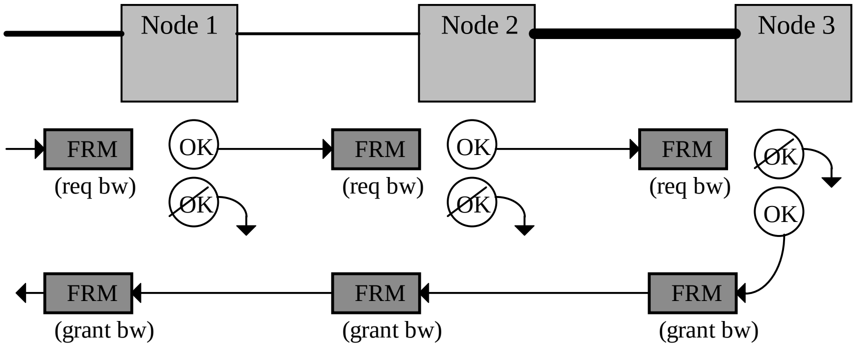


Figure 34: Fast Bandwidth Reservation

B.1.2 Routing Policies

Upon origination of a call at the user’s host, the source initiates a setup message across the UNI. This message contains details about the addresses as well as the initial set of parameters indicating the desired QoS and the source traffic descriptors. The connected ATM switch, in ATM terminology called the ingress switch, then performs two functions: it replies to the source a ‘call proceeding’ message to provide an indication that the setup request has been received and is in progress. It also initiates a routing protocol within the private or public ATM network to establish a connection with the end-system indicated in the address. The setup message is thus passed through a series of NNI to the egress switch (the ATM node to which the destination host is connected).

Finally, the setup request is forwarded through the UNI to the destination, which then either accepts or rejects the call. Notice that the route passed through several interfaces: the UNI at the ingress switch, a series of NNI within the ATM network, and the UNI at the egress switch.

This description prompts the following two questions:

1. What addresses are used to identify the different nodes in an ATM network?
2. What routing algorithm is used to forward the setup request through the ATM network?

ATM addresses differ for private and public networks. The format adopted for public networks is typically E.164, as standardized by ITU-T. Private networks use NSAP-like addresses, called ATM private network addresses (PNA) hereafter. The ATM Forum specifies an encoding scheme for E.164 addresses within PNAs to provide the interoperability of the private and public networks. Figure 35 depicts the two address formats. The PNA typically consists of an authority and format identifier (AFI), an initial domain identifier (IDI), a domain specific part (DSP) and an end-system identifier (ESI). The address format for the E.164 encoded addresses replaces and expands the initial domain identifier with the E.164 address, and reduces the size of the domain specific part. In both cases, the size of the address is 20 bytes.

AFI	IDI	DSP	ESI	SEL
(1 byte)	(2 bytes)	(10 bytes)	(6 bytes)	(1 byte)

ATM Private Network Address Format

AFI	E.164	DSP	ESI	SEL
(1 byte)	(8 bytes)	(4 bytes)	(6 bytes)	(1 byte)

Embedded E.164 Address Format

Figure 35: ATM Network Address Formats

The AFI identifies the type and format of the IDI, which represents the address allocation and administration authority. The DSP contains the actual routing information, such as routing domains or area identifiers. The ESI can be interpreted like a MAC address. It represents an actual end system.

Concerning the second question, the ATM Forum has concentrated on specifying the private-NNI (P-NNI) routing protocol. The public NNI routing functions have not yet been treated by the standards committees. The P-NNI protocol outlines virtual circuit routing, which switches the signaling request through the ATM network and therefore determines the route of the VCC’s. To ensure that the QoS indicated in the setup-request can be met with a high probability, the P-NNI protocol uses a topology state routing algorithm, similar to IGRP. Each node periodically floods a set of node and link parameters which describe the state of the network, such as the following parameters:

- Maximum Cell Transfer Delay (MCTD)
- Maximum Cell Delay Variation (MCDV)
- Maximum Cell Loss Ratio (MCLR)
- Available Cell Rate (ACR) [5]

The ATM Forum suggests both a node-by-node and a source-node routing algorithm. In the former case, each node determines the next hop to bring the setup request closer to its destination, taking the available topology state parameters into account. This algorithm has several disadvantages. First of all, it does not prevent loops. Depending on the topology state, a setup request could theoretically return to a node which has been visited before. Secondly, each node not only needs to execute its local CAC function. It also needs to evaluate the QoS across the entire network to determine the next hop, which is computationally expensive.

By contrast, in a source-node routing algorithm, the entire end-to-end path is determined by the source, according to the requested QoS and the available topology state information. The intermediary nodes then only have to perform their local CAC functions to determine whether they have sufficient capacity to support the requested call. The end-to-end path is determined using the Generic Connection Admission Control (GCAC) algorithm:

1. Examine the ACR of all links and select those links which can provide the requested ACR.
2. Execute a shortest path calculation from the source to the destination. Note that this typically results in a set of possible paths to the destination.
3. Select a path according to other link metrics, such as smallest MCTD, MCDV or MCLR.
4. If still more than one path exists, select the path with the smallest load.
5. Insert the path into the signaling request.

Note that this algorithm is based on the topology state parameters at the time the setup request is issued. This means that the algorithm does not necessarily find the optimal path from the source to the destination. Each node along the path still performs a local CAC function, because the topology state information may have changed since the path has been calculated. For this reason, the P-NNI protocol also supports the concept of crankback. For example, if the resources on an intermediary node have been exhausted but this information has not yet been flooded through the network, a source may compute a path which includes such a node. In this case, when the setup request reaches the

---

[5] Notice that the first three parameters are non-additive, whereas the last parameter is additive. Given a route consisting of several links, the overall ACR along the entire path is determined by adding the ACR's of the individual links on the path. In case of the non-additive metrics, however, the overall measure is determined by taking the smallest metric of all the links on the path. This represents the limiting link along the path.

intermediary node, the local CAC function rejects the call and hence setup fails. The source is notified of this failure and may then compute an alternative path, using by then (hopefully) up-to- date information.

B.2 Usage Parameter Control / Network Parameter Control (UPC/NPC)

The responsibility of UPC/NPC is to continuously monitor that the traffic contract is not violated while a call is in progress. The user might transmit more cells than he or she agreed to in the traffic contract, thus increasing the likelihood of network congestion. The network thus has to supervise the incoming traffic rate and compare it against the traffic rate which has been agreed to in the traffic contract. As such, UPC/NPC is essential for maintaining the QoS for all the existing calls on the network.

ATM outlines several different methods to perform this function. The most drastic option given to the network is to simply release any SVC which does not conform to the agreed transmission rates. Less drastic options are

- Cell tagging and discarding
- Traffic shaping
- Explicit flow control

Cell tagging and discarding makes use of the CLP bit in the ATM cell header (see section A.3.2). User cells may be sent with two different priorities: CLP=0 or CLP=1. A cell with CLP=1 has a lower priority and thereby a higher probability of being dropped. These priorities may be set by either the source or the network. The network continuously measures the burst size of a traffic stream using the generic cell rate algorithm (GCRA). The measured burst is compared, for example, to the PCR specified in the traffic contract. If upon arrival of a cell the PCR exceeds the measured rate, the cell is found to be conforming and accepted. Otherwise, its CLP bit is inspected. Cells with current CLP value equal to 0 are tagged, which means that their CLP bit is set to 1. This marks a cell for the remaining transmission path and therefore increases the likelihood of subsequent discarding. Cells with current CLP value equal to 1 are immediately discarded. The actual cell conformance algorithm depends on the cells service category. Section B.2.2 outlines the detailed specifications for this procedure.

Rather than violating the traffic contract and risking cells to be dropped, the source can apply traffic shaping to its stream. Shaping is a process by which the cell stream is modified in such a way that it still conforms to the traffic contract. Shaping can take several forms. The source may decide to buffer the arriving cells until they no longer violate the traffic contract as per the GCRA. This however introduces additional delays and possibly jitter. Another form of shaping is peak cell reduction, where the source deliberately transmits at a peak rate less than the one agreed to in the traffic contract. This increases the probability of conformance for cells. Yet another form of shaping is called burst length limiting, where the source limits the burst length to a level which is below the one specified in the traffic contract. Traffic shaping is defined as being optional in the ATM Forum's standard specification. The implementation of these shaping functions is thus vendor specific. The vendors may even implement additional shaping algorithms to those just described.

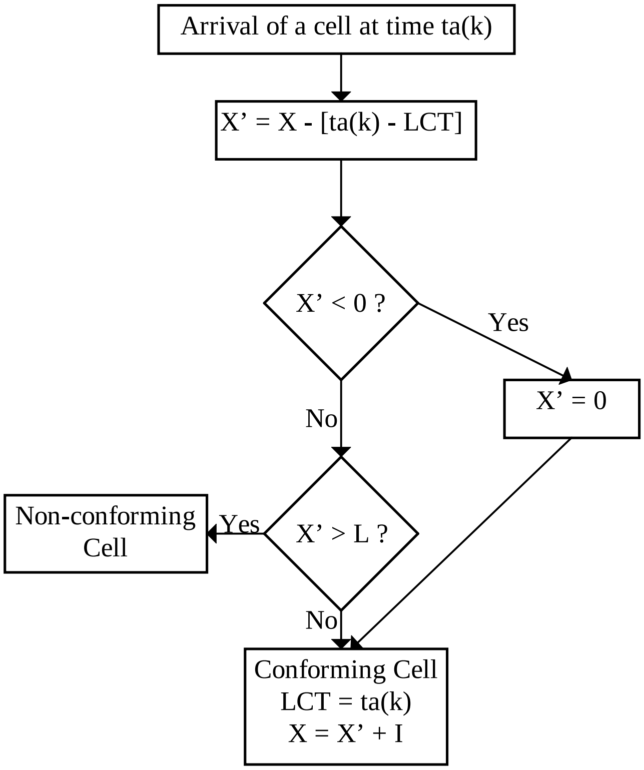
The last method for UPC/NPC listed above is explicit congestion control. This function is responsible for detecting and reacting to periods of congestion, preferably as early as possible. It uses explicit messages to inform the source of network congestion and hence throttles the transmission rate of the source. Congestion control is currently only relevant for ABR traffic. The CBR, rt-VBR and nrt-VBR service classes receive much tighter commitments from the network about the QoS requirements. The UBR traffic, on the other hand, receives no commitments from the network whatsoever, allowing the network to simply drop cells in case of heavy congestion. A sophisticated congestion control mechanism is thus not needed for this class. Section B.2.3 provides more details on this mechanism.

B.2.1 The Generic Cell Rate Algorithm (GCRA)

The UPC/NPC functions relies heavily on the generic cell rate algorithm (GCRA), also known as the leaky bucket algorithm. The GCRA defines the time-frame over which the transmission rates are measured and then compared to the values in the traffic contract. It is therefore essential for determining the compliance of a cell with the traffic contract at the UNI, and furthermore provides values for the CDVT(PCR) and the CDVT(SCR).

The purpose of the algorithm is to determine whether a cell is conforming or non conforming. It takes two parameters: an increment value I and a limit L which represent the burst size (PCR or SCR) and a tolerance value respectively. The algorithm considers a finite-capacity bucket. Each time unit, the bucket contents are drained by 1 unit. Each time a cell is conforming, the bucket content is incremented by I. If a cell arrives and the content of the bucket is less than L, the cell is conforming. If the content of the bucket is greater than L, the cell is non-conforming. This indicates that the burst size is exceeded, i.e. that the source is transmitting at a higher rate than negotiated.

Figure 36 defines this algorithm formally. It declares variables for the actual bucket counter (X) and a provisional bucket counter (X'), and keeps track of the last time that a cell has complied with the algorithm (LCT). When the first cell arrives, the bucket counter X is set to 0 and the LCT is set to the arrival time. The first value of the provisional bucket counter X' will therefore also be 0 (see diagram) and the cell is automatically conforming. The value of the bucket counter is increased by the increment I, and the last conformance time is set to the arrival time of the cell. Conformance of subsequent cells depend on the value of the provisional bucket counter X', calculated at the arrival time, as well as the limit L. Note that the value ta(k)-LCT represents the actual inter-arrival time of two cells. The value of the bucket counter X can be interpreted as the allowed inter-arrival time with respect to the measures in the traffic contract.



Key:

X	value of leaky bucket counter
X’	provisional value of leaky bucket counter
LCT	last conformance time
ta(k)	arrival time of cell k

Figure 36: The generic cell rate algorithm (GCRA)

Notice that the description of the GCRA is indeed ‘generic’. This algorithm can be used to measure compliance with respect to different parameters. In particular, it could measure the compliance of a VBR stream with both the PCR and the SCR, provided the algorithm is used twice with different parameters. The increment I takes the inverse of the PCR or SCR respectively. The limit L corresponds to the CDVT set in the traffic contract.

Furthermore, the GCRA can also be applied to different cell streams. Recall that the cells are either tagged as high or low priority in their CLP bit. The two available options defined by the ATM Forum are to either consider only those cells with a high priority. This stream is labeled CLP=0. Alternatively, all cells could be considered, regardless of their CLP value. This stream is labeled CLP=0+1. Note there is no need to decide on the conformance of the CLP=1 stream, since the cell is already tagged. The UPC/NPC functions would decide its fate - discard or transmit.

B.2.2 UPC/NPC Compliance

Now that we have explained the details of the GCRA, let us take a second look at the conformance functions specified by the ATM Forum. This section summarizes the relationship between the UPC/NPC cell tagging/discarding function and the GCRA. The conformance definition of a cell depends on two factors:

- The service class and its QoS parameters
- The CLP cell streams (CLP=0+1, CLP=0)

The ATM Forum defines different conformance rules for all possible combinations of these factors. These are outlined below.

B.2.2.1 Constant Bit Rate Conformance Definition

Recall that the QoS parameters for CBR traffic are PCR and CDVT(PCR). This implies two versions of conformance definitions: one considering only the CLP=0+1 stream and one considering the CLP=0+1 and CLP=0 streams.

The first version measures the bursts of all cells using the GCRA with respect to the PCR. If a cell is found to be conforming, irrespective of the value of its CLP bit, it is transmitted. Otherwise, the cell is discarded immediately. This version of the conformance definition does not allow tagging of cells. Figure 37 illustrates this policy.

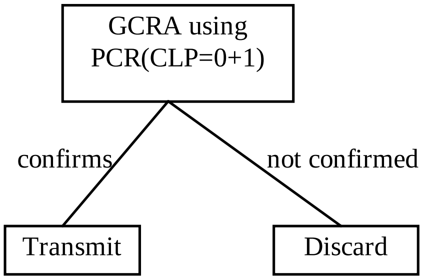


Figure 37: Conformance Definition for CBR traffic using PCR(CLP=0+1)

The second version used two GCRAs: one for the CLP=0+1 stream and one for the CLP=0 stream. The first measures the conformance with respect to the PCR of all cells (i.e. CLP=0+1). The second GCRA measures the conformance of cells with respect to the PCR of only the CLP=0 cells. The UPC/NCP function is outlined in Figure 38.

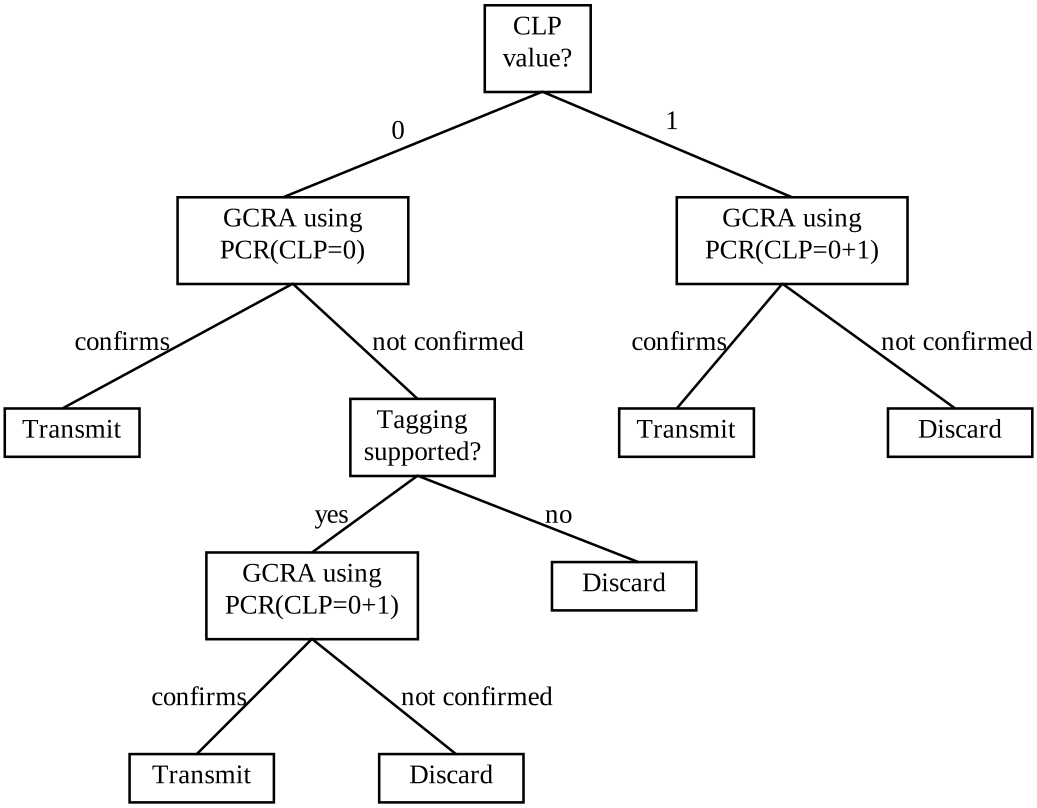


Figure 38: Conformance Definition for CBR traffic using PCR(CLP=0+1) and PCR(CLP=0)

Note that the conformance definition version used for a particular traffic stream is defined in the source traffic descriptor of the traffic contract. Also, in order for a cell to be tagged, the source has to request tagging and the network has to support it.

B.2.2.2 Variable Bit Rate Conformance Definition

Recall that the QoS parameters for VBR traffic are PCR, SCR, CDVT(PCR) and CDVT(SCR). This implies two versions of conformance definitions: one considering only the PCR CLP=0+1 stream in conjunction with the SCR CLP=0 stream, and one considering the PCR CLP=0+1 in conjunction with the SCR CLP=0 stream.

The UPC/NCP functions are outlined in Figure 39 for the first version and in Figure 40 for the second version.

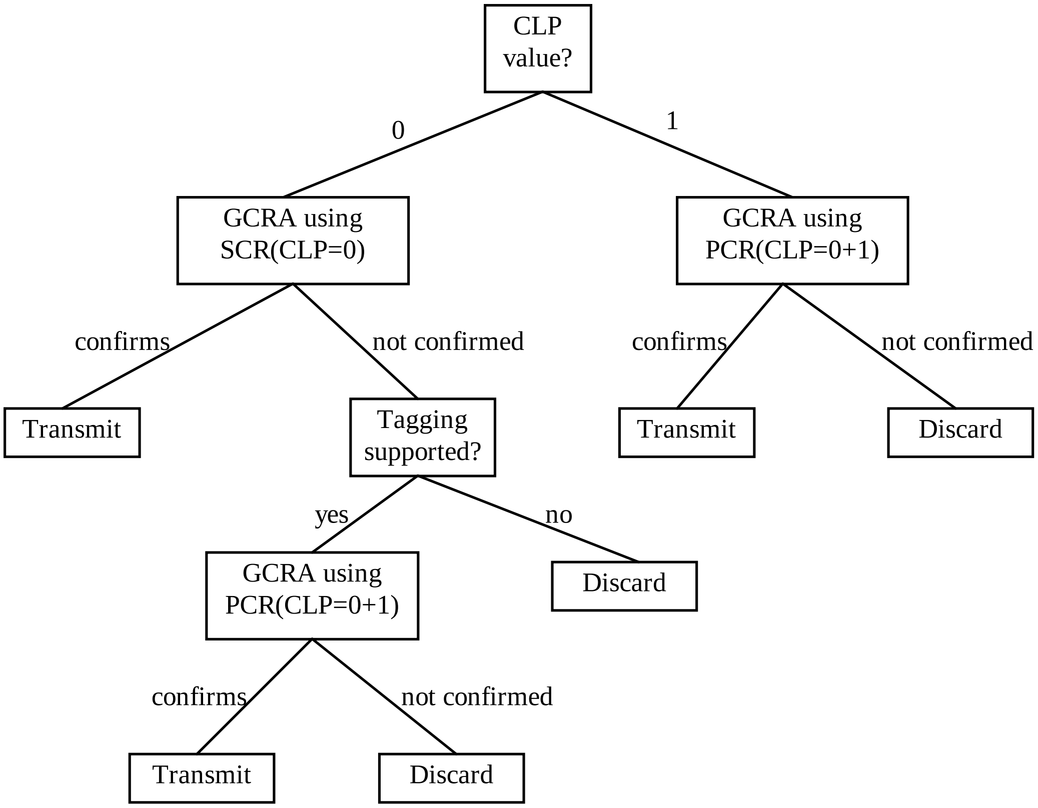


Figure 39: Conformance Definition for VBR traffic using PCR(CLP=0+1) and SCR(CLP=0)

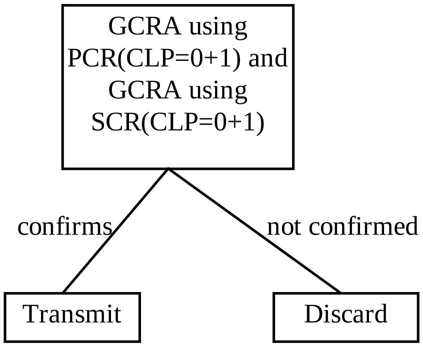


Figure 40: Conformance Definition for VBR traffic using PCR(CLP=0+1) and SCR(CLP=0+1)

Like above, the conformance definition version used for a particular traffic stream is defined in the source traffic descriptor of the traffic contract, and in order for a cell to be tagged, the source has to request tagging and the network has to support it. Note that for the second version, tagging is not defined by the standard.

B.2.2.3 Available Bit Rate Conformance Definition

The conformance definition for this service class is slightly more complicated. Only CLP=0 cells are considered in the ATM Forum’s specification. The actual conformance definition is network specific. The minimal conformance definition requires one GCRA with the PCR for the CLP=0 stream as its parameter. The algorithm has to take the flow and congestion control mechanisms specified for ABR services into account. Loosely speaking, an ABR cell is only considered as non-conforming if ‘its arrival time and those of the preceding cells on the connections could not have resulted from the ideal transmission times of an ABR source’[ATM TM 4.0]. Strictly speaking, the UPC/NPC functions are replaced by explicit congestion control mechanisms which force the source to adjust to network congestion. Discarding cells is very unlikely unless the congestion control mechanisms are ineffective. Section B.2.3 provides more detail on ABR congestion control mechanisms.

B.2.2.4 Unspecified Bit Rate Conformance Definition

The ATM Forum leaves the conformance definition for UBR service up to the network providers. Recall that the traffic descriptors for this class are PCR and CDVT(PCR). These may not be enforced by a network provider.

B.2.3 ABR Congestion Control Schemes

Two forms of congestion control are currently under discussion:

- Credit-based congestion control
- Rate-based congestion control

Under the credit-based congestion control scheme, each link at each switch maintains a certain credit balance of the available buffer at the downstream node. These credits can be alternatively described as buffer-tokens of the downstream node. Before a cell can be transmitted, a buffer-token or credit has to be available. Upon transmission, the credit balance is reduced. As soon as the downstream node forwards a cell itself, freeing its buffer space, a notification is sent to the upstream node to increase its credit balance. If no buffer-tokens are available at a node, the transmission process is paused. This scheme is illustrated in Figure 41.

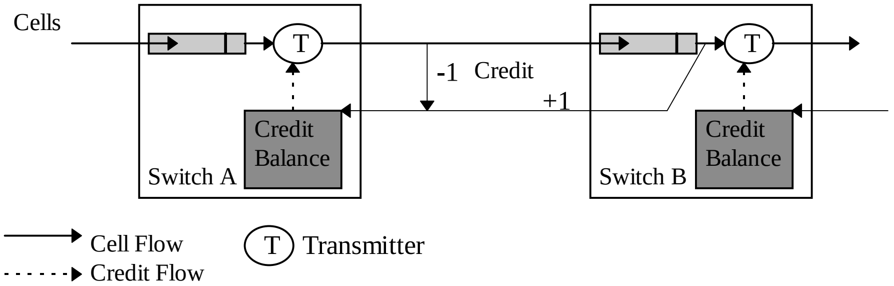


Figure 41: Credit-based Congestion Control

One of the disadvantages of credit based congestion control is that it is very expensive to implement. The switches need to maintain a separate FIFO queue per port as well as a scheduler to determine which queue has cells and credits available for transmission. Furthermore, the switches need to have a mechanism to recover from losing credits. In the extreme case of losing all credits for a particular port, transmission across a particular link would come to a halt. These complications make credit-based flow control very expensive to implement, which is why this scheme has been put on hold by the ATM Forum.

Whereas the above scheme represents a permanent (or proactive) congestion control scheme, rate based congestion control is only activated when switches actually become congested (i.e. a reactive scheme). The network provides the sources explicitly with congestion information in form of special management cells. Three different variations of rate-based congestion control are under discussion: Forward Explicit Congestion Notification (FECN), Backward Explicit Congestion Notification (BECN) and Explicit Rate Control (ERC).

The FECN scheme makes use of the explicit forward congestion indication (EFCI) field in the ATM cell header. If a switch becomes congested, it marks the EFCI field in the cell header and forwards the cell to the destination. Upon arrival of a marked cell at the destination, a notification process is started which transmits a congestion notification cell back to the source. As soon as the source receives this congestion notification cell it throttles the transmission of the cell stream, thereby reducing congestion in the network. A disadvantage of the FECN scheme is the transmission delay introduced by first sending the congestion information to the destination and then returning it to the source. With large round-trip delays, the end-to-end delay of the congestion notification cell could be too long for this scheme to work effectively.

The BECN scheme overcomes this problem as follows. Instead of marking the EFCI field of a cell in the downstream direction, it directly sends a resource management cell to the source announcing the congestion. This has the advantage of reducing (not alleviating) the notification delay as compared to FECN. However, this congestion control scheme requires more complicated hardware at the switches to detect and filter the congestion state in order to communicate the necessary information to the source.

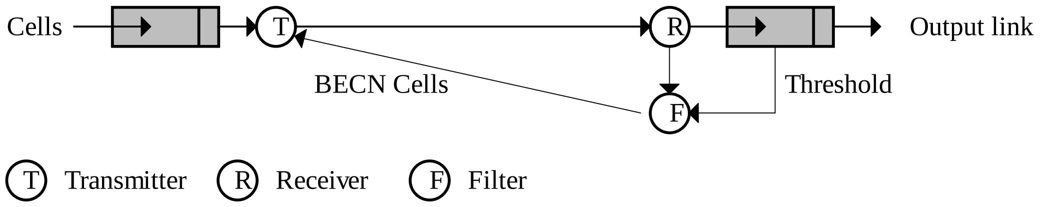


Figure 42: Rate-based Congestion Control using BECN

The third rate-based congestion control scheme, ERC, uses a similar principle to BECN. Instead of notifying the destination about the congestion level, a switch in the network may send a resource management cell to the source, indicating the required rate at which the source should transmit. The difference between BECN and ERC lies in the frequency with which this information is provided to the source. The former only sends cells when and during a congestion period. As soon as the congestion at the switch has been alleviated, the transmission of BECN cells stops. With ERC, the source is continuously informed by the network at which rate it is supposed to transmit, irrespective of the congestion level of individual switches. This scheme is therefore more proactive, whereas BECN and FECN could be interpreted as reactive congestion control schemes.

All of the three rate-based congestion control schemes have the advantage of being easier to implement than the credit-based scheme. Also, the congestion feedback from the network is automated and event-based. The traffic sources dynamically adjust to new transmission rates as a result of this feedback. For this reason they are seen to be more appropriate for a WAN ATM network. However, the rate-based congestion control schemes currently only cover point-to-point connections, not multipoint connections, and they are not covered in the ATM Forum's UNI signaling standard.

Considering a WAN environment supporting high-speed fibre optic links, congestion control schemes might fail altogether. All of these schemes require the buffering of cells at intermediate switches to avoid cell loss. With high-speed links and a large round-trip delay, the amount of space required to buffer these cell is equal to the bandwidth-delay product. Providing this buffer space might be too expensive. Also, accommodating congestion control schemes introduces an additional processing delay at the switches. With tight delay constraints of real-time applications in particular, implementing an acceptable processing delay at the switches might be prohibitively expensive. Due to these problems with existing congestion control methods, the ATM Forum currently discusses an alternative scheme, called Quality Flow Control.

The ATM Forum's specification adopts the ERC method for ABR traffic. In order to fully implement this method, a further set of parameters is required in the traffic contract for ABR. As shown in section A.2.1 in Table 21, the traffic descriptors for ABR are a PCR, a MCR and a CDVT/PCR. In addition to those, the following parameters are used by the network to control the traffic rate of ABR streams:

- Initial Cell Rate (ICR): this is the start-up rate at which the source is allowed to transmit.
- Additive Increase Rate (AIR): this is the additive increment of the transmission rate for the source.
- Number of resource management cells (Nrm): this specifies the number of user cells that are sent before a resource management (RM) cell is transmitted to the destination.
- Allocation of bandwidth for resource management cells (Mrm): this parameter controls the allocation of bandwidth between the RM cells and the data cells.
- Rate Decrease Factor (RDF): this factor determines the decrease in the transmission rate. The parameter is multiplicative, so the smaller the value, the larger the reduction in the transmission rate.
- Allowed Cell Rate (ACR): this parameter is self-explanatory.
- Number of resource management cells before rate decrease (Xrm): this parameter specifies the number of RM cells which may be in transit before the source has to decrease its transmission rate.
- Time-out Factor (TOF): this controls the maximum time allowed between the transmission of an RM cell and the required reduction in the transmission rate.
- Xrm Decrease Factor (XDF): this factor allows a dynamic modification of the Xrm

These parameters enforce the following ABR transmission operation:

The source start off with an initial transmission rate of ICR. It transmits at a rate of ACR, which may never exceed the PCR nor be below the MCR. Every Nrm user cells, a resource management cell is sent to the destination. If a RM cell returns and indicates congestion in the network, the source has to reduce its transmission rate by at least ACR*Nrm/RDF, provided that this reduction does not violate the MCR. If it does, then the transmission rate is set to MCR. Similarly, if a RM cell returns and does not indicate congestion, the source may increase the ACR by AIR*Nrm, provided that the PCR is not violated. Otherwise, the ACR is set to the PCR. If Xrm RM cells have been sent out, but none of them have returned, the source has to reduce its transmission by ACR*XDF, provided that the MCR is not violated. If it is, then the ACR would be set to the MCR.

B.3 Management Services

We will now outline the management functions of ATM based broadband networks located in the layer management plane of the B-ISDN protocol reference model. In this section we first introduce the concept of Operations, Administration and Maintenance Cells (OAM cells) and discuss the performance management functions. We will then describe the fault management functions of ATM networks in detail.

As the name implies, Performance Management (PM) is responsible for maintaining an adequate performance level of an ATM network. Its main function is to calculate and provide vital performance statistics. Not only do these give the network manager an indication about the current performance of the ATM layer services. They are also essential to accommodate the QoS concept. In order to negotiate a QoS during CAC, the network needs to have an indication about the current values of the QoS parameters (delay, jitter, loss statistics etc.), as outlined in section A.1.1. Based on these values the CAC algorithm can then negotiate a QoS level for the new call and fix the negotiated values in the traffic contract. A further use of the statistics obtained through PM is then to supervise the QoS parameters of the existing traffic contracts, assuring that the network itself does not violate the agreed upon parameters.

Fault Management (FM) encompasses four different functions to prevent or react to network failures. These four functions are as follows:

- continuity check
- cell loopback
- remote defect indication
- alarm indication

The first two functions in this list operate preventatively, trying to detect network errors as soon as possible. The last two functions react to network failures and provide a sufficient means for recovery.

Before outlining the PM and FM procedures in detail we introduce the concept of OAM cells, which are essential for providing ATM layer management functions.

B.3.1 OAM Cells and their Flows

OAM cells are ATM cells with a special format to provide the management functions of ATM networks. They can typically be categorized in two dimension: the connection dimension and the functionality dimension.

The connection dimension classifies the OAM cells according to their flow through the network. OAM cells may flow at the VPC level or the VCC level, referred to in the ATM standard as F4 and F5 respectively. For both of these levels the OAM cells flow either end-to-end or per segment as indicated in Figure 43. The VCI and PT fields in the ATM cell header are used to indicate this dimension.

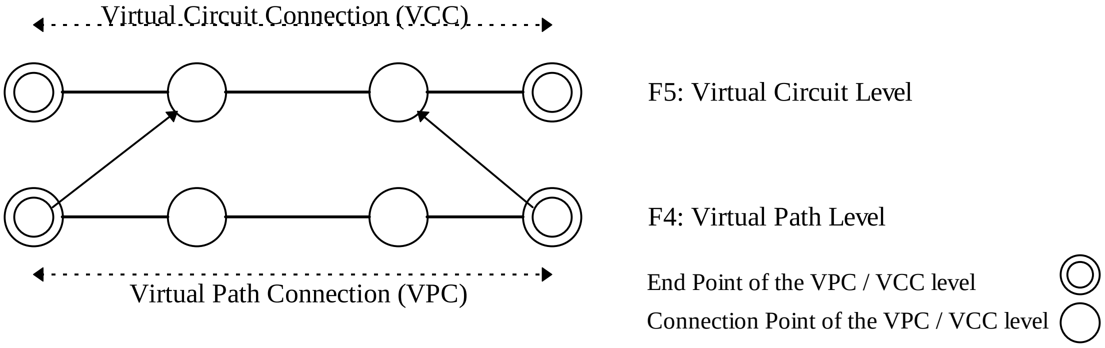
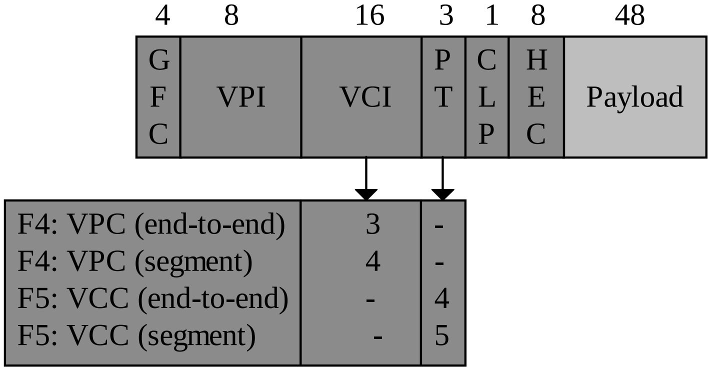


Figure 43: VPC and VCC Levels

End-to-end operation flows refer to OAM cells which are used to communicate operations information across an entire end-to-end VPC (or VCC). They may only be inserted and removed at the endpoints of the VCC / VPC, but allow all the intermediary nodes to inspect the carried OAM information. The end-to-end OAM cell at the VCC level is given a PT value of 5, leaving the VCI field untouched (this is needed to uniquely identify to which VCC the management information pertains). The end-to-end OAM cell at the VPC level is given VCI value 4, not using the PT field in the header.

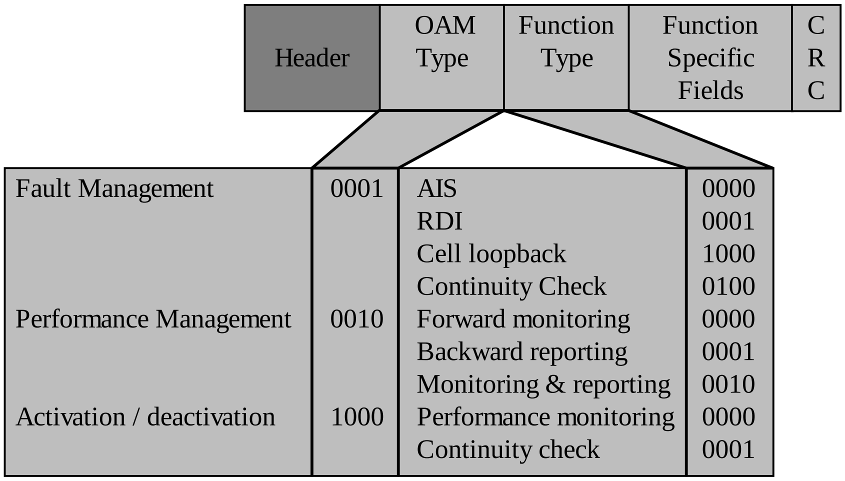
Those OAM cells flowing at the segment level carry operations information per segment. In this case, intermediary nodes may not only inspect the cell contents, they may also insert such an OAM cell. Like before, only the endpoint of the VPC or VCC is allowed to terminate the OAM cell. The segment OAM cell at the VCC level is given a PT value of 4, again leaving the VCI field untouched. The segment OAM cell at the VPC level is given VCI value 3, again not using the PT field. Figure 44 summarizes the different combinations of values for this dimension.



F4: VPC (end-to-end)	3	-
F4: VPC (segment)	4	-
F5: VCC (end-to-end)	-	4
F5: VCC (segment)	-	5

Figure 44: OAM Cell Formats in the connection dimension

In the second dimension, the OAM cells are classified by function. The distinction here is made between performance management, fault management or activation / deactivation OAM cells. To provide this distinction the fields labeled ‘OAM type’ and ‘Function type’ in the OAM payload are used. The former field takes the bit-value ‘0001’ for fault management, the bit-value ‘0010’ for performance management and the bit-value ‘1000’ for activation / deactivation. Each of these classes is further subdivided into several modes, for which the latter field is used. Performance monitoring operates in the tree modes: forward monitoring, backward reporting and monitoring and reporting, indicating in which direction the PM OAM flows[6]. The ‘Function Type’ is set to the bit-values ‘0000’, ‘0001’ and ‘0010’ respectively. For fault management four different modes are defined: alarm indication signal, remote defect indicator, cell loopback and continuity check. These four modes as well as their values in the field ‘Function type’ will be explained in detail below. Finally, the activation / deactivation OAM cell is used to turn the management functions on and off. ‘Function Type’ here takes the value ‘0000’ to toggle the activation mode of performance management, and the value ‘0001’ to toggle the activation mode of fault management. This mode provides an important flexibility to ATM network management, as will be shown in the next section. Figure 45 summarizes the functional dimension of the different OAM cell formats.



Fault Management	0001	AIS	0000
		RDI	0001
		Cell loopback	1000
		Continuity Check	0100
Performance Management	0010	Forward monitoring	0000
		Backward reporting	0001
		Monitoring & reporting	0010
Activation / deactivation	1000	Performance monitoring	0000
		Continuity check	0001

Figure 45: OAM Cell Formats in the functionality dimension

---

[6] forward: from VPC/VCC starting point to VPC/VCC end point, backward: from VPC/VCC end point to VPC/VCC starting point, monitoring and reporting: both directions.

B.3.2 Performance Management Procedure

The performance management procedure calculates performance statistics about the network which are either used by the CAC function to decide whether to accept or reject a newly arriving call, or to monitor the QoS of existing calls. The following statistics are calculated:

- Cell loss ratio (CLR)
- Cell misinsertion rate (CMR)
- Cell error ratio (CER)
- Severely errored cell block ratio (SECBR)
- Cell transfer delay (CTD)
- Mean cell transfer delay (Mean CTD)
- Cell delay variation (jitter, CDV))

The starting point of the VPC / VCC inserts a forward monitoring PM OAM over a block of user cells, containing a time stamp, an error detection code for the block of user cells as well as the size of the block. The ATM Forum currently standardizes the block size to either 128, 256, 512 or 1024 user cells with a variability of 50%[7]. This OAM cell is transmitted to the end point of the VPC / VCC. Upon arrival, the cell is extracted and the statistics are calculated according to the following formulae:

- $\text{CLR} = \dfrac{\text{Lost Cells}}{\text{Total Transmitted Cells}}$

The VPC / VCC end point here simply looks at how many cells have arrived in the last block as compared to the number indicated in the OAM cell. This value is typically in the range between 10^{-1} to 10^{-15}.

- $\text{CMR} = \dfrac{\text{Misinserted Cells}}{\text{Time Interval}}$

Misinserted cells here are cells that are mistakenly forwarded along the wrong VPC / VCC because of an error in the cell header. Severely errored cell blocks should be ignored in this calculation according to the ATM Forum’s specification.

- $\text{CER} = \dfrac{\text{Errored Cells}}{\text{Successfully Transferred Cells} + \text{Errored Cells}}$

A cell is considered as errored if the cell header is corrupted or if the payload type is different than indicated in the PT field.

---

[7] This variability provides flexibility to the performance management function. With CBR applications for example, strict time delays have to be met, which must not be violated by the insertion of PM OAM cells.

- $\text{SECBR} = \dfrac{\text{Severely Errored Cell Bocks}}{\text{Total Transmitted Cell Blocks}}$

A cell block is considered as severely errored if a certain number of cells in that block have been misinserted, errored or even lost.

- CTD = ‘arrival time’ - ‘departure time’

Arrival time is considered here the point in time when the OAM cell arrives at the end point of the VPC VCC. Departure time is considered to be the time indicated in the time stamp field of the OAM cell, thus the time that the cell has been transmitted by the start point of the VPC / VCC.

- Mean CTD = average of CTD’s

This statistics simply takes a number of CTD’s from different OAM cells which have arrived at the end point of the VPC / VCC and computes an average.

- CDV = variance of CTC’s

Similar to the Mean CTD, except that here the variance is calculated instead of an average.

After having computed these statistics, the end point of the VPC / VCC connection then optionally returns the result to the start point of the VPC / VCC connection using a backward reporting PM OAM cell. Note that all the intermediary nodes along the VPC / VCC are able to inspect these cells too.

To obtain statistically meaningful estimates of the above parameters, a number values should be considered. This means that a considerable number of OAM cells have to be transmitted before an accurate estimate can be obtained. This might turn out to be a problem, in particular when considering SVCs. If for example, an SVC with a short duration is considered and a block size of 1024 is considered, then only a few OAM cells might be generated by the PM procedure, leading to an insufficient number of estimates for the above parameters. Another problem in this procedure is the sheer number of VC’s which are typically set up in an ATM network. If for all of these VC’s PM is enabled, then the total number of OAM cells generated might be considerable, thus impacting on the operations and efficiency of the network. Instead of measuring the performance of the network, PM would then actually reduce the performance.

B.3.3 Fault Management

The fault management functions are intended to handle any error situations resulting from actual hardware / software failures, corrupted VPI / VCI translation tables or the network's inability to delineate ATM cells from the payload at the underlying physical layer. In addition to these reactive measures, two proactive functions are provided: loopback testing and continuity checking. The former is a service for network managers to test the connectivity of network connections. The latter is intended to distinguish a discontinued connection from an idle connection.

B.3.3.1 Fault Detection

The following procedure has been devised in the ATM Forum's standard to deal with cases where a network component has actually failed or an intermediary nodes VPC / VCI translation tables have become corrupted. Figure 46 is used to illustrate the concepts.

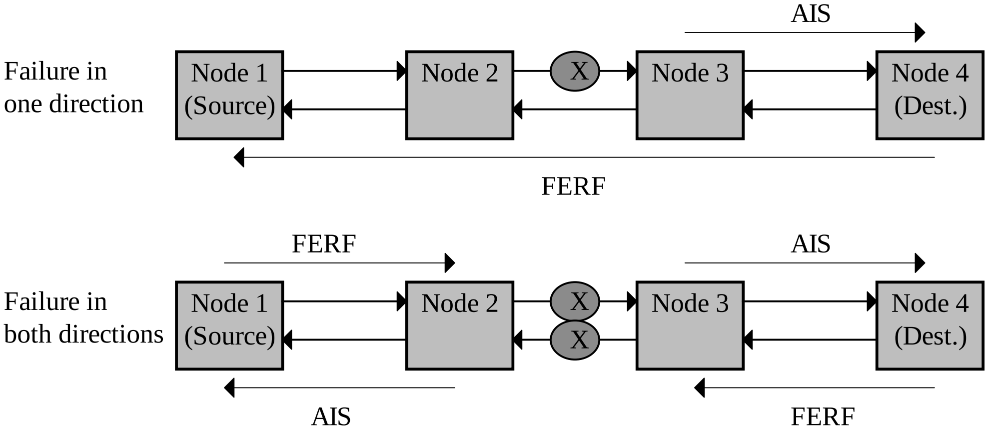


Figure 46: Failure detection and reporting using AIS and FERF

The node downstream from the failure is the first one to detect the instability in the system. Before raising an alarm, a delay is incurred to allow the underlying physical layer to recover automatically (the instability might be caused by the physical layer and not be severe). However, in case that the instability retains, the node generates an AIS cell and transmits it to the end point of the VPC / VCC. The field 'Function Type' in the payload takes the value '0000' for such a cell. In addition to this, the failure location is reported. As soon as the AIS cell reaches the end point of the VPC / VCC, another FM OAM cell is generated and sent to the start point of the VPC / VCC. This cell is called a FERF (Far End Reporting Failure)[8] cell in the standard and has a 'Function Type' value of '0001'. Upon arrival of the FERF cell at the start point of the VPC / VCC the relevant recovery actions are then initiated.

---

[8] Also called a RDI (Remote Defect Indicator) cell.

Note that the node detecting the failure keeps on generating AIS cells to the end point of the VPC / VCC periodically in the order of seconds until the failure is corrected.

B.3.3.2 Loopback Control

The intention of loopback control is to provide network managers with the capability to verify the connectivity of the network, isolate network faults or perform measurements of the cell delay. The field ‘Function Type’ in the payload takes the value ‘0001’ for FM OAM loopback cells.

The procedure to execute a loopback is defined as follows: any node along a VPC / VCC can insert the FM OAM loopback cell, specifying any other node on the same VPC / VCC as a loopback point. The field ‘Loopback Location’ is used for this purpose. Upon arrival of the cell at the loopback point, the field ‘Loopback Indication’ is changed to indicate that the cell indeed has reached the loopback point. The relevant node then returns the OAM cell and the inserting node terminates the cell. Notice that this procedure implies that the loopback does not necessarily have to be initiated and terminated at the start / end point of a VPC / VCC. It could be inserted and looped back anywhere along a VPC / VCC, as illustrated in Figure 47. As a result of this flexibility, several FM OAM loopback cells can co-exist along the same VPC / VCC. To allow the inserting node to identify its own loopback cells the field labeled ‘Correlation Tag’ is used.

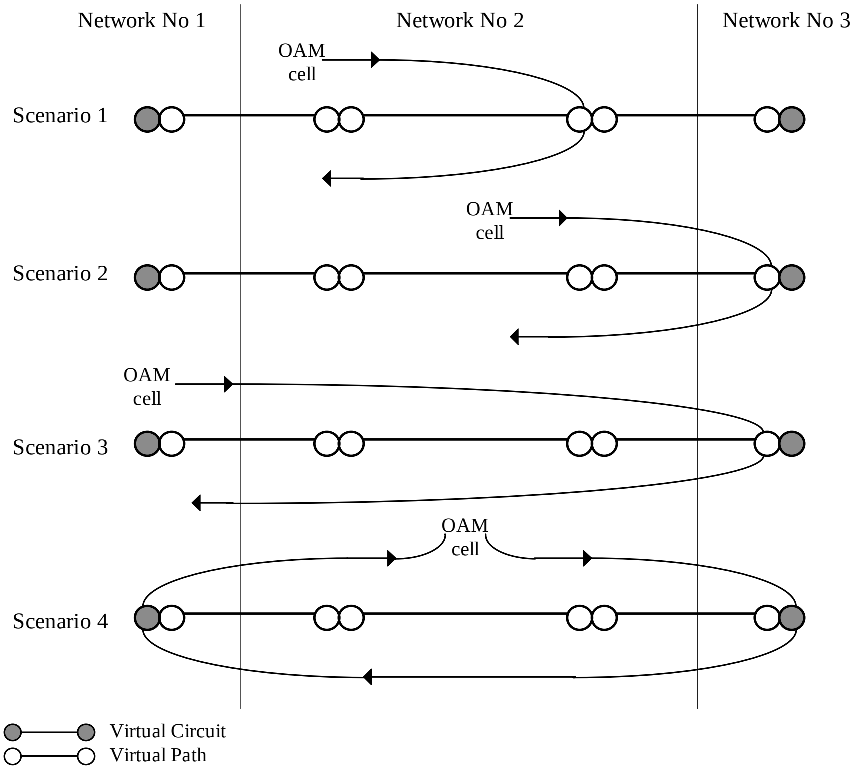


Figure 47: Example loopback scenarios

B.3.3.3 Continuity Checking

The last fault management function currently standardized is continuity checking. Its purpose it to detect possible VPC failures. In particular, it is intended to distinguish a failed VPC from an idle VPC. If no cells flow along a VPC, is this a result of a failed VPC or because no traffic is presently being transmitted? The FM OAM cells with ‘Function Type’ ‘0100’ is used for this purpose, and periodically transmitted between the two end points of a VPC. If the downstream end point does not receive such a cell within a certain time period, the VPC connection is assumed to be faulty and a FERF procedure is generated. The start point of the VPC can then initiate any corrective actions.

Notice that this function is only performed at the VPC level, not the VCC level, in the current ATM Forum standards. A possible explanation of this is the overhead associated with OAM cells. A VPC bundles a group of VCCs together, hence reducing the number of continuity checking cells which have to be generated.

Glossary and Acronyms:

The terms indicated with an asterisk (*) are taken from the Fore System's home page.

ABR	Available Bit Rate traffic. Class of Service in ATM which can make use of the available bandwidth, typically dynamically adjusted by the network. Priority is given to CBR and VBR service classes.
ATM Forum	Group of companies who intend to provide and use ATM networks. The ATM Forum's technical committees work on a common standard for ATM. It is the principal organizations body to foster development of ATM technology.
ATM subnetwork	ATM network owned and managed by a single operator. This could be a public carrier, or a private organization. The subnetwork could be an ATM-based LAN or WAN.
ATM Switching System	A node switching ATM cells. Part of an ATM network.
ATM*	Asynchronous Transfer Mode. A form of digital transmission based on the transfer of units of information known as cell. It is suitable for the transmission of image, voice, video and data.
AAL*	ATM Adaptation Layer. The AAL translates digital voice, image and video and data signals into the ATM cell format and vice versa. Five AALs are defined.
BECN	Backward Explicit Congestion Notification. Form of congestion control where the congested node sends a cell to the source of the traffic informing it about the congestion. The source can then reduce its transmission rate.
B-ICI	Broadband inter carrier interface. Interface between two ATM network carriers.
B-ISDN	Broadband Integrated Services Digital Network.
B-ISSI	Broadband inter switching system interface. Interface between two ATM switching systems.
Broadband	Bandwidth above 2 Mbps.
CAC	Connection Admission Control. Functions executed to determine whether a newly arriving call can be admitted to the network or not. This is typically a function of the network's available resources.
CBR	Constant Bit Rate traffic. Traffic which is sent at a constant rate, as opposed to bursty traffic.
CDV*	Cell Delay Variation. A measurement of the allowable variation in delay between the reception of one cell and the next. (Usually expressed in thousandths of a second, or milliseconds (msec.). Important in the transmission of voice and video traffic, CDV measurements determine whether or not cells are arriving at the far end too late to reconstruct a valid packet.
CDVT	Cell Delay Variation Tolerance. See also CDV
CLR	Cell Loss Ratio. Ratio of dropped / lost cells to the number of transmitted cells.

Congestion Control	Functions to regulate the congestion of the network devices, in particular the overflow of buffers. For example: credit-based congestion control, rate-based congestion control.
CRC	Cyclic Redundancy Check. An error detecting code in which the code is the remainder resulting from dividing the bits to be checked by a predetermined binary number [14]
Control Plane	Plane in the B-ISDN Protocol Reference model describing the services provided by different layers to perform control functions, such as CAC.
Destination	Network node where traffic terminates.
ERC	Explicit Rate Control. A form of congestion control where the source receives information from the network about how much traffic it can currently transmit. This information is generated continuously, not just upon congestion.
ETSI	European Telecommunication Standards Institute.
FECN	Forward Explicit Congestion Notification. Congestion control method whereby the congested node first notifies the destination of a VPC / VCC about its congestion. The destination then generates a congestion notification cell to the source.
FRM	Fast Resource Management. Management function whereby a resource management cell is transmitted through the network from source to destination prior to the transmission of data. This cell contains the resource requirements of the call. The nodes either reserve the required resources or reject the cell, in which case the call setup process fails.
GCRA	Generic Cell Rate Algorithm, also known as the leaky bucket algorithm. See Appendix C for a description.
HEC*	Header Error Control. An 8-bit Cyclic Redundancy Code (CRC) computed on all fields in an ATM header, capable of detecting single bit and certain multiple bit errors. HEC us used by the Physical Layer for cell delineation.
IETF	Internet Engineering Task Force. A standards organization.
ILMI*	Interim Local Management Interface. The standard that specifies the use of the Simple Network Management Protocol (SNMP) and an ATM management information base (MIB) to provide network status and configuration information.
Management Plane	Plane in the B-ISDN Protocol Reference model describing the services which are provided to the network for management.
MBS	Maximum Burst Size. The maximum rate at which traffic can be sent at once.
MCR	Minimum Cell Rate
MIB	Management Information Base. Database containing vital management information about the network. Contains the current utilization figures (e.g. for VPCs / VCCs) and other statistical information.
NMS	Network Management System. System which is responsible for the management of the network. Provides vital information and functions to the network manager, as described in this paper.

NNI*	Network to network interface. The interface between one ATM switch and another, or an ATM switch and a public ATM switching system.
nrt-VBR	non-real-time Variable Bit Rate traffic
OAM cells	Operations, Administration and Maintenance Cells. Cells used within the ATM network to perform the network management functions.
PCR	Peak Cell Rate. Traffic descriptor indicating the maximum transmission rate of cells from the user.
PDU*	Protocol Data Unit. A unit of information (e.g. packet or frame) exchanged between peer layers in a network.
PT	Payload Type. Special field in the ATM cell header indicating the type of information carried as payload. A 3-bit descriptor in an ATM cell header that indicates whether the cell is a user cell or a management cell.
PTI	see PT.
PVC*	Permanent Virtual Circuit. A generic term for any permanent, provisioned communications medium.
QoS*	Quality of Service. The ATM Forum has outlined five categories of performance (Classes 1 through 5) and recommends that ATM's quality of service should be comparable to that of standard digital connections.
QFC	Quality Flow Control. An alternative flow control scheme currently discussed by the ATM Forum.
rt-VBR	real-time Variable Bit Rate traffic
SCR	Sustainable Cell Rate. A traffic descriptor, specifying the maximum cell transmission rate.
SNMP	Simple Network Management Protocol. A protocol used for the management of communications networks. Not only used for ATM networks, but also for other types of networks.
Source	Network node where traffic originates.
SVC	Switched Virtual Circuit. Virtual Circuit switched at arrival time of a call. A generic term for any switched communications medium.
Traffic contract	Contract between the user and the network operator determining the Quality of the Service. The user commits to transferring at a certain traffic rate. Similarly, the network will commit to transferring the traffic rate.
Traffic Descriptors	Parameters describing the traffic profile of the user, such as PCR, SCR, MBS and MCR.
UBR	Unspecified Bit Rate traffic.
UME	UNI Management Entity
UNI	User Network Interface. Point where the user accesses the ATM network.
UPC*	Usage Parameter Control. The function of ATM network equipment that controls the Cell Loss Priority bit to control congestion on the network.
VCC*	Virtual Circuit Connection. A logical communications medium identified by a VCI and carried within a VPC. VCCs may be permanent virtual channel connections (PVCCs), switched virtual channel connections (SVCCs) or smart permanent virtual channel connections (SPVCCs). Further, VCC is an end-to-end logical communications medium. Another acronym, VCL (virtual channel link) is more precise, referring to the

	single segment object identified by a VCI and carried within a VCP. Similarly, a VPC is an end-to-end object and a Virtual Path Link (VPL) is identified a VPI within a link.
VCI*	The field in the ATM cell header that labels (identifies) a particular virtual channel.
VC*	Virtual Channel Identifier. A generic term for any logical communications medium. NOTE: VC does not stand for virtual channel. Virtual channels are referred to as VCCs (Virtual Channel Connections). There are three classes of VC's: permanent, switched and smart (or soft) permanent.
VPC*	Virtual Path Connection. A logical communications medium in ATM identified by a virtual path identifier (VPI) and carried within a link. VPCs may be permanent virtual path connections (PVPCs), switched virtual path connections (SVPCs) or smart permanent virtual path connections (SPVPCs). VPCs are uni-directional.
VPI*	Virtual Path Identifier. The field in the ATM cell header that labels (identifies) a particular virtual path.