Traffic Performance in an ATM network

Magnus Jonnerby

99-05-14

MSc Thesis

Department of Teleinformatics, Royal Institute of Technology

Ericsson Telecom AB, Datacom Networks & IP Services

Department of Teleinformatics, Royal Institute of Technology

Examiner: Gunnar Karlsson

Ericsson Telecom AB, Datacom Networks & IP Services

Supervisor: Jörgen Axell

Abstract

A stand-alone demonstrator of a traffic performance monitoring (TPM) tool has been implemented. It monitors real-time bandwidth statistics to support a service provider to reach higher bandwidth utilization in a network with a number of virtual private networks (VPNs), achieved by over-allocation in the links. Statistical multiplexing makes it possible to make use of the sum of allocated but currently not utilized bandwidth in the connections. This means in reality that functions similar to those in the TPM tool would support the service provider to sell more bandwidth than the actual capacity of a physical link. However, this introduces an estimated risk of violating the connections' quality of service (QoS). The TPM tool monitors bandwidth statistics on link- (physical and logical) and connection (VP and VC) levels. The graphical user interface (GUI) of the TPM tool is divided into a physical- and a logical network view. This makes it possible to distinguish the information and the statistics between different network levels, e.g the physical network and a customer's VPN. Two possible ways to over-allocate are static over-allocation and dynamic CAC functions. They are mentioned but not closely investigated in this report. The integration of the TPM tool into a real management system is not considered in this work.

Table of Contents

1.0 Introduction

1.1 Project description

1.1.1 Background

An ATM network that is used as a multi-service network, integrates different services like voice, music, telephony and video to run over the same network. Traffic performance monitoring in the management system assists the operator to effectively operate and maintain the network. A main goal for the operator is to manage the network in a cost effective way and still retain the services' quality of service (QoS). This project is focused on the *monitoring of traffic performance statistics* in the management system. The project is performed at the business unit Datacom Networks & IP Services of Ericsson Telecom AB.

1.1.2 Goals

Ericsson's ATM switching system (AXD301) contains an extensive set of functions for performance monitoring. This project aims at showing how some of these functions should be used by a management system. A first step is to come up with new ideas and suggestions of what traffic performance statistics an operator would like to monitor in a management system, and then to produce a proposal of how these statistics can be presented in the management system's user interface. Based on the proposal a stand-alone demonstrator of a traffic performance monitoring (TPM) tool has been implemented.

Primary goals for the project:

- Suggest what traffic performance statistics an operator may want from an ATM network.
- Make a proposal on how the traffic performance statistics can be presented to the operator.
- Implement a demonstrator of the graphical user interface (GUI) for the proposed TPM functions.

1.2 Report outline

The report is organized as follows, chapter 1 to 4 gives a general theoretical background for the project area, chapter 5 to 8 describes the implementation of the proposed TPM functions and chapter 9 summarizes the work. Readers already familiar with network management may skip chapter 1 to 4.

Theoretical background (1 - 4)

Chapter 2 introduces a network management system and describes the general architecture. The five functional areas of network management suggested by ISO are described.

Chapter 3 is an introduction to performance management in ATM networks. It explains the difference between quality of service (QoS) and network performance (NP) and gives a short description of how to monitor performance statistics in a management system.

Chapter 4 presents the basic concept of switching in ATM networks and explains the architecture of the AXD301 switching system. It ends with a brief description of the AXD301 management system (AMS).

Implementation part (5 - 8)

Chapter 5 gives a background and presents a scenario for the implementation of the traffic performance monitoring (TPM) tool.

Chapter 6 describes the modelling of the TPM tool.

Chapter 7 gives a brief description of the implementation of the TPM tool.

Chapter 8 contains evaluation and discussion around the TPM tool. Summarizes possible problems when implementing a TPM tool into a real management system.

Summary

Chapter 9 summarizes and makes conclusions of this work. It suggests some possible future works.

Appendix

Complementary reading lists the material used in the prestudy of this work but not directly referenced in the report.

2.0 Network management system (NMS)

2.1 Introduction

A *network management system* (NMS) is a set of software functions to help optimize the operation and maintenance of a network. The NMS is usually located in a central management center from where the network operator (referred as just operator) controls, monitors and configures the network elements.

Network management has become a key issue for many companies, for which the use of data communication services constantly increases. The companies should carefully select a data communication solution that guarantees their required *quality of service* (QoS). *Service providers* sell data communication services to customers and relieve the customer from the quandary of all network management tasks.

The primary goal for an NMS is to operate and maintain the network in a *cost effective* way without violating the services' predefined QoS. It is also important to keep the operation and administration of the NMS as *simple* as possible for the operator.

Network management includes many tasks. ISO has made a conceptual model which divides network management into five functional areas: *Fault-, Configuration-, Accounting-, Performance-* and *Security* management.

2.1.1 Service provider

The use of data communication services constantly increases and the QoS demands are getting more critical. For many companies and organizations, running a network is not considered as a main core activity, so the market is rapidly growing for service providers.

The service provider's primary goal is to manage its network in a cost effective way and still fulfil the customers' *service-level agreements* (SLAs). A good NMS monitors the network resources for the operator, so they can be used in an effective way and simplify the administration and maintenance.

A service provider is competitive, if it is flexible to rapidly adapt to new market demands. The network needs an intelligent and flexible infrastructure, which easily can be adjusted for network growth and new services.

2.2 General NMS architecture

Most network management architectures have the same basic structure and set of relationship even if the components can be named differently. Here are some basic components found in most of the NMSs (figure 1):

- *Manager*
- *Management station*

- *Management Information Base (MIB)*
- *Management agent (agent)*
- *Network management protocol*
- *Network device (device)*

The *Manager* is the central part of the NMS. It handles the polling, which is about getting data objects from the MIBs residing in the network devices. These objects (information) are stored in the manager database.

A *Management station* is a computer workstation, which is the interface for the operator to operate and maintain the network. From the management station the operator can remotely supervise, monitor and configure individual network devices. Usually a separate computer workstation is used for just operator tasks, but sometimes a common computer is used for running the manager and the operator tasks.

A *Management Information Base* (MIB) is a database with a collection of objects. These objects are logical representations of resources of the device. There is a MIB on each device, which usually is designed in accordance with some approved standard.

A *Management agent* (agent) is a software packet residing on each device, it administrates the MIB and handles the “communication” with the manager and other devices in the network.

A *Network management protocol* is used by the manager and the agents to communicate with each other. Well known network management protocols are Simple network management protocol (SNMP) [2], and Common management information protocol (CMIP)[3].

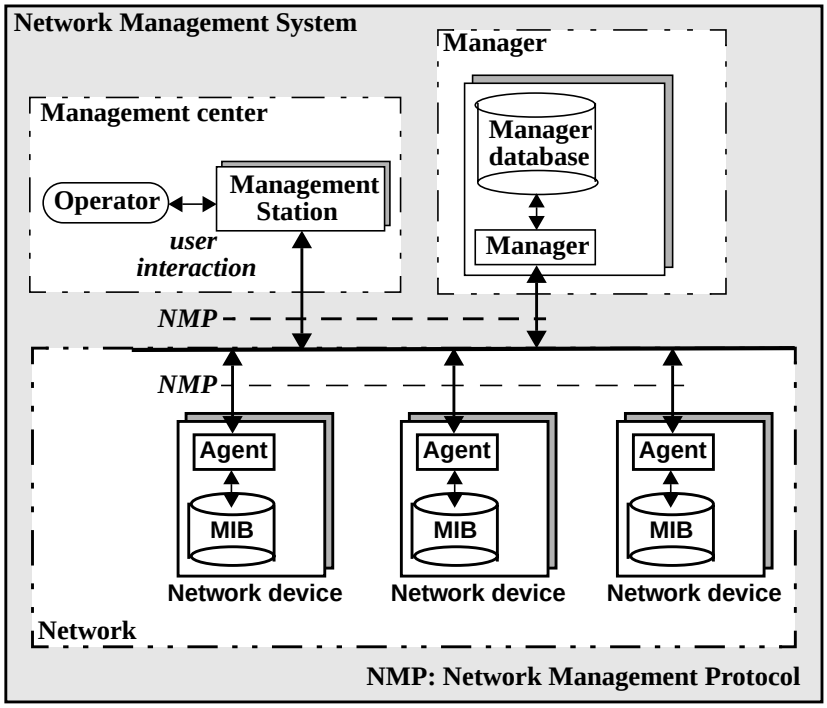


Figure 1. Conceptual model of a network management system.

2.3 Network management goals

An NMS should help the operator to operate and maintain the network in a cost effective way. This in a dynamic and constantly evolving environment for the NMS. Simplicity to administer network updates is another important characteristic. Here are some primary goals for the NMS.

- *Maximizing availability*: The network is available when it delivers services to the user with the right QoS.
- *Enabling adaptability*: The environment in which the NMS is used (e.g. service providers, companies, organizations, etc.) can rapidly change its structure by business agreements. It should be possible to adapt the NMS to the new structure.
- *Easy to change services and the QoS*: The NMS integrates many services over the same network. Services and users requirements of QoS evolve in time, so it should be easy to change or add new services and adjust their QoS.
- *Easy network upgrade*: Multi-service networks are built of many network elements. The NMS should support network components from many vendors, so it become easy to expand or upgrade the network. Market demands from the operators usually cause vendors to cooperate. Most of the vendors use open technologies and try to support the same standards.

2.4 Network management areas

The International Organization for Standardization (ISO) has suggested to divide network management into five functional areas [4].

- ***Fault management***: Detection, identification and correction of faults in network components.
- ***Configuration management***: Operation and maintenance of individual network elements and monitoring of configuration data for individual network elements.
- ***Accounting management***: Measure user utilization of network resources and services. Resource utilization is used to tune the network and service utilization is used for billing customers.
- ***Performance management***: Monitoring performance of individual network elements and the whole network. Common performance parameters are e.g. availability, utilization, throughput and response times.
- ***Security management***: Protect traffic from unauthorized access, administrating network access, monitoring and logging access to sensitive network resources.

Some of the tasks may overlap many functional areas. The model should primarily ease the understanding of NMS functions. Vendors of NMS usually have their systems divided into modules similar to those areas suggested by ISO.

2.4.1 Fault management

The primary goal of fault management is to detect, log and fix network problems in order to minimize network downtime and make the network run efficiently. Avoiding faults in the network by preventive handling is the best fault management. In case a fault still occurs, the fault should fast be detected, identified, isolated, and handled. The fault and the management actions should be logged. Fault management involves several steps:

1. *fault detection*: Detect that a fault has occurred in the network and identify problem symptoms.
2. *identification and isolation*: Identify the network element where a fault has occurred. Isolate the problem and take the faulty network element out of operation.
3. *fault measures*: Fix the problem.
4. *test the fault measures*: Test if the fault measures really fixed the problem and that they do not have any negative side effects to the rest of the network. Then put it on operation as usual.
5. *logging*: Record the fault detection and the management actions that were used to solve the problem.

Efficient fault management is very critical in an NMS. It costs a lot of money if the network is down. Service providers have signed service-level agreements (SLAs) with their customers. The service provider loses potential revenues when its services are not available to the customers. Violations of the customers' SLAs are usually connected with a

charge for the service provider. For other companies, network downtime may result in serious loss in productivity.

2.4.2 Configuration management

Configuration management is a corner stone for the other functional areas. It is difficult to manage a network, to not say impossible, if there are no functions to configure network elements. Primary goals for configuration management are monitoring and reconfiguring of the hardware- and software settings of the network elements. An operator should be able to monitor and reconfigure individual network elements from the management center.

Monitoring and configuration of network elements cover areas like:

- *software*: Monitor current software, software version and software settings in network elements. Possible from the management center to upgrade the software and reconfigure the software settings in a network element.
- *hardware*: Monitor current hardwares and hardware settings in the network.
- *traffic*: In an ATM network there are a lot of parameters to adjust which affect the traffic performance. It is desirable to avoid congestion and find an optimal performance tuning. Some tuning options to mention are the connection admission control (CAC), usage parameter control (UPC), traffic shaping, priority control, buffer settings, etc. These notions are described later in the report.

2.4.3 Accounting management

Accounting management measures individual users' utilization of network resources and services. Examples of network resources are e.g. switches, computers and connections. Information about resource utilization is used to reach fair access between users and optimal usage of the network. A service provider uses the service utilization to bill the customers. For example, a service provider could need to measure the usage time of each service class for a customer.

2.4.4 Performance management

Performance management measures various aspects of network performance. which in this report is distinguished into two areas: traffic performance and signalling performance. The main focus of this work is on the traffic performance. Here is a simplified description of both areas:

- *traffic performance*: How effective is the network in forwarding data.
- *signalling performance*: How effective is the "communication" between network elements.

Network operators and administrators use parameters of network performance to reach optimal network utilization. These parameters can describe the performance in individual network elements or part of the network. This internal performance in the network is not of interest to an end-user, which is only interested in the QoS (more about this in section 3.2.1).

The manager in the NMS polls performance parameters from each network node and stores them in a common database. Collection of the parameters are done by a network management protocol of which the Simple network management protocol (SNMP) is most common to use. The operator monitors and analyzes the performance statistics in the management center to discover bottlenecks and poor resource utilization, then do some reconfigurations if necessary. Performance statistics are important for planning network growth or upgrade.

2.4.5 Security management

The goals of security management are to *protect network resources* from intentional or unintentional damage and to *protect sensitive data* sent over the network from unauthorized access. A usual practical solution is to partition networks and networks resources into subareas with different security levels. A security policy decides if a user is authorized for a specific subarea. Key areas for security management are:

- Mapping which network resources and subareas a user is authorized to use.
- Administration of access to sensitive networks resources, thus checking if the user is authorized to use the network resource.
- Logging and monitoring access to sensitive network resources.

2.5 Standardizations bodies.

Today's multi-service networks are built of many network elements. The operators want vendors to use a common *standardized interface* for network elements, which makes it possible for the operator to choose between many vendors when the network need to be upgraded or expanded. It is important that the vendors build network elements on *open technologies* and support the same *standards*. The business environment in which the NMS is used can rapidly change in structure, so it is important that network elements and NMS from different vendors can coexist in the new environment.

Standardizations bodies

Here are some important organizations working with standardizations issues in the NMS, ATM and Internet areas:

- *ITU - T* (International Telecommunication Union - Telecommunication Standardization Sector)
- *ATM Forum*
- *IETF*- The Internet Engineering Task Force

ITU - T is known as the world's leading telecommunication standards authority. Technical standards for ATM can be found amongst its I - and Q - series recommendations for broadband-ISDN (B-ISDN).

ATM Forum was started by four manufacturers, Nortel, Sprint, Sun Microsystems and Digital Equipment Corporation in 1991. The goal was to speed up the standardization process of ATM. ATM Forum is divided into three main areas:

- *technical committee*: Technical standards are developed by the principal members of the forum.
- *marketing*: The market awareness and education (MA & E) committee is responsible for delivering educational materials to the general telecommunication market to increase the knowledge of the ATM technology, and improve awareness of the ATM capabilities.
- *user need analysis*: The end user roundtable (ENR) committee is reserved only for user members. Should give an understanding of users' need from ATM.

*IETF i*s an international open community of network designers, operators, vendors and researchers in the evolvement of the Internet. IETF is the primary standards body for setting Internet standards. The technical work is divided by topic into several working groups where each group is managed by an Area Director (AD). Work in progress are published as Internet-drafts which are not approved standard documents. A Internet-draft is only valid for a limited time. When a specification document is approved as standard, the work is published as an *RFC* (Request for Comments).

There is a slight different between the work in ITU - T and the ATM Forum. The ATM Forum prepares technical specifications for e.g. different interfaces in the network, while ITU - T is more concerned with general standards and general models, where the implementation perspective is not specified.

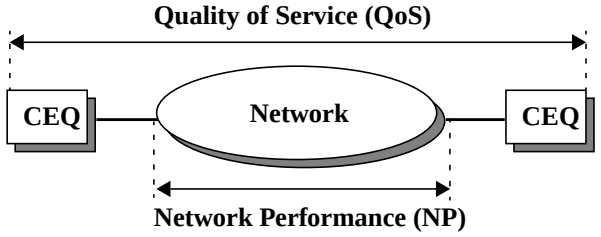
3.0 Performance management

3.1 Introduction

Performance management is about measuring various aspects of network performance and active tuning of the network. ITU - T Recommendation I.350 [5], classifies network performance in two categories:

- ***Quality of Service (QoS)*** describe the user-oriented performance parameters. It is a quality measure of how well the network supports the services over the network from an end-user perspective.
- ***Network performance (NP)*** parameters measure the network efficiency and the effectiveness in individual network elements. These efficiency-oriented parameters are aimed for the operator.

Some common performance parameters are: availability and response time (which are *user-oriented*), throughput and utilization (which are *efficiency-oriented*).



CEQ: Customer equipment

Figure 2. The relationship between quality of service and network performance.

A service provider agrees on a *service-level agreement* (SLA) with a customer. The SLA specifies the QoS the service provider should deliver to the customer. The service provider uses the NP parameters to achieve cost effective network administration and operation. It must be able to verify the customer's QoS in its NMS.

3.2 Network performance categories

In ITU-T recommendation I.350 [5], some QoS and NP parameters are defined.

3.2.1 Quality of Service (QoS)

QoS parameters help an end-user to verify that services are delivered by the network with a certain quality. ITU - T defines QoS as follows, "Collective effect of service per-

formance which determine the degree of satisfaction of a user of the service" [5]. QoS is characterized by the combined aspects of service support and service operability performance. It also includes the user's subjective degree of satisfaction but in this report it is restricted to effects possible to measure at end-users' service access points.

A user is not concerned with how the network is operated and managed or any aspects of internal network performance. The user is interested in the end-to-end service performance. This can be described by a set of QoS parameters, which should have the following *characteristics*:

- Focus on end-user perceivable effects rather than their causes within the network.
- Definitions should be independent of network architecture.
- Measurable between service access points.
- Independent of NMS, thus the parameters should have the identical meaning in different NMSs.

It is important that a service provider is able to measure QoS parameters for each service it delivers to a customer, so it can verify the customer's SLA.

3.2.2 Network performance (NP)

Network performance parameters are of interest to the operator and other persons involved in the technical aspects of the network. NP is divided into the categories:

- *traffic performance:* How effective the network is in sending data packets. Examples of such parameters are throughput, number of discarded-, errored- and lost cells. This information helps the operator to discover bottlenecks and tune the network for better performance.
- *signalling performance*: How effective the network elements are "communicating" with each other, e.g. during the connection establishment.

NP parameters are used for the *purpose* of:

- *analyzing network performance*: They help the operator to reach optimal resource utilization and cost effective operation and maintenance of the network.
- *strategic network growth planning*: NP statistics are important when operators plan for network expansion or upgrade.
- *preventive handling*: Good preventive handling minimizes downtime in the network.

NP parameters should have following *characteristics*:

- Independent of end-user equipment, thus equipment used to connect services to the network are not of interest.
- Measurable at boundaries of network elements.
- Information for system development, network planning, operation and maintenance, thus the parameters should have a clear relation to the technical architecture of the network and individual network elements.

3.3 Performance parameters in ATM networks

3.3.1 QoS parameters in ATM networks

Traffic parameters describe traffic characteristics of a source and are grouped into source- and connection traffic descriptors. This described terminology are used by ATM Forum [19]. *Source traffic descriptors* are used during the connection establishment and *connection traffic descriptors* specify the characteristics of an ATM connection. In this report the terminology traffic parameters is used for both source and connection traffic descriptors. Here are some important traffic parameters for *bandwidth* requirements in an ATM network [19, 6]:

- *Peak cell rate* (PCR): Maximum bandwidth the connection is allowed to generate.
- *Sustainable cell rate* (SCR): Average traffic bandwidth the connection is allowed to generate.
- *Minimum cell rate* (MCR): Demands on minimum available bandwidth for the connection.
- *Maximum burst size (MBS)*: Maximum allowed traffic burst size, when bandwidth is PCR.

QoS parameters for requirements on *delay* in an ATM network [19, 6]:

- *Maximum Cell Transfer Delay* (maxCTD): Maximum allowed difference between reception and transmission time for a cell between two end-user points.
- *Peak-to-peak Cell Delay Variation* (peak-to-peak CDV): Maximum allowed difference between the maxCTD and minCTD for a cell between two end-user points. The minCTD represents the minimum transfer time for a cell.

Traffic contract

Before a user can send traffic over an ATM network a connection must be set up, thus the user needs to sign a *traffic contract* (service contract) with the network. The user presents traffic- and QoS parameters for the network. If the network can serve the user's traffic demands, then a traffic contract is signed (figure 3). The set of actions taken by the network for negotiating traffic contracts is called *connection admission control* (CAC). The CAC functions only allow a new connection if it not threatening the QoS of existing connections.

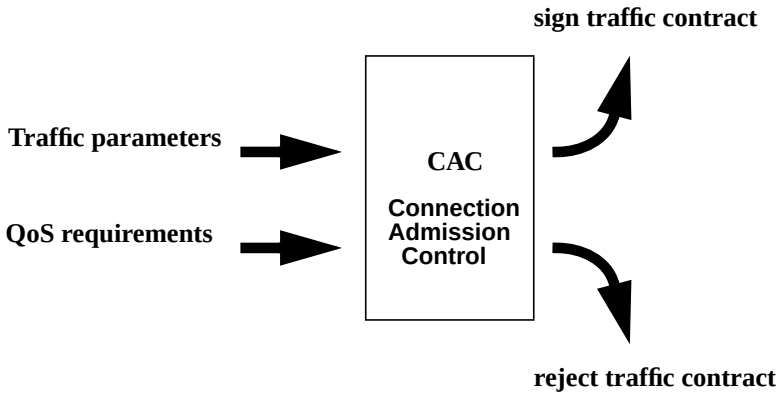


Figure 3. Principle for negotiating a traffic contract

Service categories

An ATM network can carry many types of applications. ITU and ATM Forum has defined five service categories with different traffic characteristics. Here are the *service categories* according to ATM Forum's definitions[19, 6]:

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

Constant bit rate (CBR) supports user applications that transmit at a fixed bandwidth. Parameter used for signing the traffic contract is PCR.

Real-time variable bit rate (rt-VBR) is designed to support variable bandwidth connections with low delay requirement. Parameters used for signing the traffic contract are PCR, SCR and MBS.

Non-real-time variable bit rate (nrt-VBR) is designed to support variable bandwidth connections without any requirements on the delay. Parameters used for signing traffic contract are PCR, SCR, MBS. Cells transferred within the traffic contract should expect low cell loss ratio.

Unspecified bit rate (UBR) is a "best effort" service class with the flow control left to user layers. When UBR connections get congested, then some cells in the buffers are just thrown away. UBR does not give any service guarantees. Parameter for signing the traffic contract is PCR. This value may or may not be used by the CAC and UPC procedures, depending of the implementation.

Available bit rate (ABR) has flow control and shares the available bandwidth left after the CBR and VBR categories have been served. Parameters for signing the traffic contract are PCR and MCR. It guarantees low cell loss ratio and a minimum bandwidth specified by the MCR, if the user adapts its traffic to the flow control.

Traffic parameters	CBR	rt_VBR	nrt_VBR	UBR	ABR
PCR	specified	specified	specified	specified (1)	specified (2)
SCR, MBS	not specified	specified	not specified	not specified	not specified
MCR	not specified	not specified	not specified	not specified	specified

Notes:

1: May not be used by the CAC and UPC procedures

2: Specifies the maximum rate at which the ABR source may ever send.

Figure 4. Traffic parameters for the ATM service categories.

3.3.2 NP parameters in ATM networks

ITU-T recommendation I.356 [7], defines speed, accuracy and dependability performance parameters for cell transfer in the ATM layer. QoS parameters are the total effect of performance of three layers: Physical layer, ATM layer and ATM Adaptation layer (AAL).

ITU-T defines NP parameters derived from a set of possible cell transfer outcomes. A *cell transfer outcome* is based on the observation of a cell between two separated measurement points (send and receive points) under a specified time (Tmax). Here follows descriptions of the possible cell transfer outcomes. A transmitted cell is either *successfully* transferred, *errored* or *lost*. A received cell is *misinserted* when no corresponding cell has been transmitted, this can e.g. occur as a result of errors in the cell header. A cell block is a sequence of N (arbitrary positive integer) cells transmitted consecutively on a given connection. When more than M (arbitrary positive integer, but $M < N$) of the received cells within a cell block is either errored, lost or misinserted, the block is *severely errored*. The values of N and M are set in the implementation. For closer descriptions of the cell transfer outcomes, see [7].

An NP parameter is estimated from detection of cell transfer outcomes during a period of time (T) at the measurement points (MPs). They are located at interfaces where the ATM layer is accessible, e.g. interfaces of an ATM network and an end-user equipment.

ITU-T defines the following NP parameters [7]:

- *Cell error ratio* (CER)
- *Cell loss ratio* (CLR)
- *Cell misinsertion rate* (CMR)
- *Severely errored cell block ratio* (SECBR)

- *Cell transfer delay* (CTD)
- *Cell delay variation* (CDV)

Cell error ratio (CER) is the ratio of the total number of (n.o.) *errored cells* to the total n.o. *transferred cells*. Cells contained in severely errored cell blocks (SECBs) should be excluded from the calculation.

Cell loss ratio (CLR) is the ratio of the total n.o. *lost cells* to the total n.o *transferred cells*. Cells contained in SECBs should be excluded from the calculation.

Cell misinsertion rate (CMR) is the number of *misinserted cells* per time unit. Cells contained in SECBs should be excluded from calculation.

Severely errored cell block ratio (SECBR) is the ratio of the total n.o. *SECBs* to the total n.o *cell blocks*.

Cell transfer delay (CTD) is the difference between reception and transmission times for the cell. *Mean cell transfer delay* is the arithmetic average of a specified number of CTDs.

Cell delay variation (CDV) is associated with two parameters:

- *1-point CDV* describes the variations of cell arrivals in one MP. Network queues and buffering procedures between the source and the MP affect the value.
- *2-point CDV* is based on observation of cell arrivals at two MPs that delimit a virtual connection portion. It gives e.g. indication of queues within the connection portion.

3.4 Performance monitoring

Before the operator can monitor performance data in the network management center, the data must be fetched from the network nodes. Some critical design issues of the performance monitoring are:

- *division of workload:* What analysis and preparation work of performance data should be done in the remote network node, and what should be done in the central NMS.
- *data collection:* What performance data is necessary to achieve efficient performance management. Fetching much redundant data from the network nodes would generate a lot of traffic and waste bandwidth.
- *polling interval:* Efficient performance management requires that the performance data is not too old. Selection of polling interval is a trade-off between generating much traffic and processor workload and getting out-dated performance data.

Remote monitoring MIB (RMON) with SNMP is a common architecture chosen by many NMS vendors. RMON is a MIB specification that makes it possible to gather e.g. traffic statistics from the RMON agents in the network nodes (see figure 5). Conventional MIBs are not designed to store traffic statistics.

3.4.1 SNMP and RMON

The data structure of conventional MIBs do not support any statistical parameters for analyzing the traffic performance. Using SNMP to regularly poll the network nodes consumes considerable bandwidth. RMON was designed to provide proactive monitoring and diagnostics for distributed LAN networks.

The performance data (statistics) can be categorized into two groups:

- *Real-time statistics* are gathered with SNMP polling of the RMON MIBs. These statistics are often critical to the performance management and supports the operator to actively tune network performance. Therefore network nodes need to be polled regularly.
- *Historical statistics* are generally a set of values that are evaluated or just aggregated during a period of time in the network node. The historical statistics can be fetched from the network node in bulks with FTP. FTP is more efficient to use than SNMP for bulk data transfer. These statistics are important to e.g plan network growth.

3.4.2 RMON

RMON is described in RFC 1757 [8], and specifies remote monitoring of network devices. It can monitor traffic statistics for the two lowest layers in the OSI model. RMON2 extends RMON to monitor traffic statistics in the upper layers (3-7 in the OSI model). Note that RMON2 does not replace RMON, it is complementary. Here are definitions of components used with RMON (figure 5):

- *RMON agent* is an SNMP agent that can be monitored from a remote RMON manager.
- *RMON probe* (probe) is the combination of the software agent and the hardware on the network device on which it resides. An RMON probe can be stand-alone or embedded in the network device.

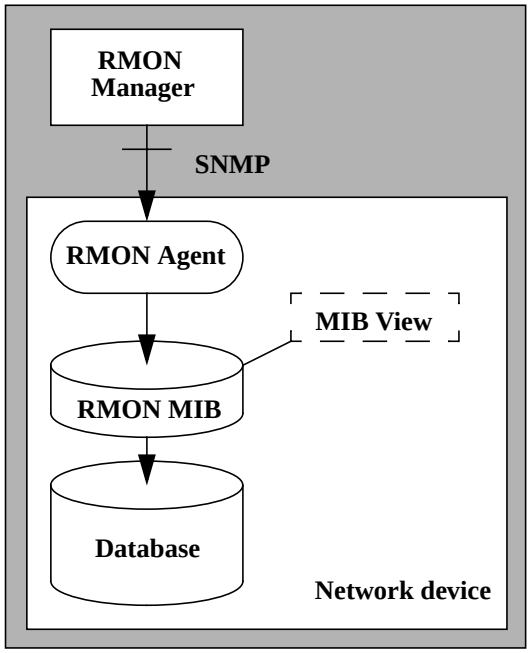


Figure 5. Principle of an RMON device

The purpose of RMON is to do much of the work, e.g. collection of statistics and diagnostics in the network device instead of in the NMS. It requires some extra computational work in the network device, but reduces the SNMP traffic and the processing load in the NMS. RMON introduces some level of intelligence to the network device by diagnostic and fault detection.

Five RMON goals are stated for remote network management [8]:

- *off-line operation:* The probe in the remote monitoring device continuously performs diagnostics and collects statistics even when there is no contact between the management station and the network device. A network failure or an intentional attempt to reduce traffic can be the reason for broken contact. When the probe detects an exceptional condition it tries to notify the management station.
- *proactive monitoring:* The monitor should continuously run diagnostics and log network performance. When detecting a failure, it should notify the management station and store statistics of the failure. Historical data can be important in analyzing the causes of failure. It should be possible for the management station after a failure to monitor statistics of the failure.
- *problem detection and reporting:* The monitor can be configured to recognize an error condition. When an error condition is detected, the monitor should log the event and notify the management station.

- *value added data:* The data collected in the RMON device can be used for further purposes than just management functions It can expand and refine the functions in the NMS, e.g. resource utilization per user
- *multiple managers:* The monitor should manage to be accessed by multiple management stations, potentially concurrently.

3.4.3 RMON extensions for ATM networks

The RMON MIB provides statistics and management functions for Ethernet and Token Ring. Adapting RMON to ATM networks require some design changes of the MIB and extended functionality, ATM Forum has made contributions on this topic [21]. Special problems for implementing RMON in ATM networks are e.g. the high speeds, cells vs. frames issues and the connection-oriented nature of ATM. Here are some important design issues for adapting RMON to an ATM switch [21]:

- *the placement of the probe*: The probe can be stand-alone or embedded, and the cells can be monitored directly or by copying.
- *virtual connection nature*: A traditional RMON probe collects statistics per port, a probe in an ATM switch requires to monitor statistics per virtual connection.
- *data reduction mechanism*: The high speeds and complex collection requirements of the probe within the switch make it necessary to reduce both the agent and the NMS for processing cell-traffic statistics.

Placement of the probe

There are four possible ways to attach the probe to the ATM switch (figure 6):

- A stand-alone probe attached to a single port, the cells are copied to the probe.
- An embedded probe within the switch, the probe has no access to the switch fabric so the cells are copied to the probe.
- An embedded probe within the switch with access to the switch fabric, the cells are monitored directly without copying.
- A stand-alone probe, tapping the NNI link between two switches. The cells are monitored directly without copying.

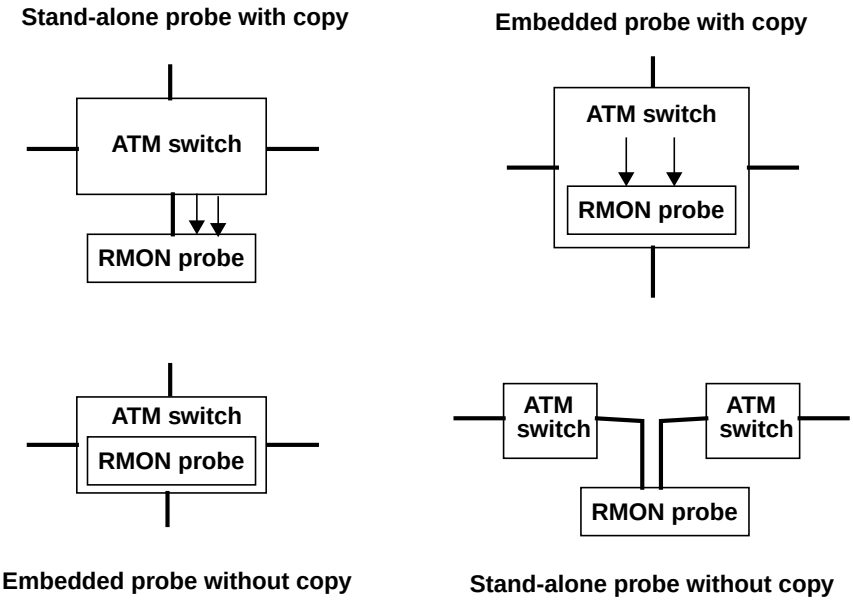


Figure 6. The placement of the RMON probe.

A stand-alone probe will not load the switch processor, but embedded probes are cheaper than stand-alone probes.

Virtual connection nature

The NMS needs to configure and monitor statistics per virtual connection instead of per port, as in traditionally RMON. For example, it is necessary to be able to define a VP as a single MIB object with a mechanism that aggregates the traffic from all the VCs that belong to the same VP into a single collection.

Data reduction mechanism

It costs a lot of processing capacity for the probe to maintain a traditional RMON MIB. With new MIB objects and added functionality for adapting RMON to ATM networks it is necessary to reduce the probe and the NMS from processing data. Generally, the most effective way to reduce processing in the probe is to reduce the amount of collected data. It is called *pre-collection data reduction* and an example of such a realization is statistical sampling. Further, the MIB tables should be designed to reduce the overall redundant data collection, thus minimize the duplication of data in different MIB tables. Pre-collection data reduction alone is not enough, it is also necessary to reduce the SNMP transactions to update the MIB tables in the NMS, this is called *post-collection data reduction.* A mechanism to achieve post-collection data reduction is e.g. collection aggregation, which gives the possibility to control the amount of data presented in the MIB tables.

ATM-RMON MIB

ATM Forum specifies a new MIB called *ATM-RMON MIB* to extend RMON for ATM networks [21]. This MIB design is not described in this report, but the interested reader can read the document “Remote Monitoring MIB Extensions for ATM Networks" (see ref. 21) on its own.

4.0 ATM switching

4.1 Introduction to ATM switching

ATM connections

ATM is connection oriented; a virtual connection is set up between two end-points before data is transferred. There are two types of virtual connections, *virtual path* (VP) and *virtual channel* (VC). A VP is a bundle of VCs which have common end-points but different physical paths through the network. The ATM cell header contains a virtual path identifier (VPI) and a virtual channel identifier (VCI); here-in they are referred to the common name "connection identifiers".

ATM switching

Cells are propagated along a virtual connection through the ATM network, passing through a number of switches. The main function of the ATM switch is to relay cells from an input port to a proper output port and set appropriate connection identifiers of the outgoing cells. A switching table maps the input ports to the output ports and translates the connection identifiers between incoming and outgoing cells. A goal for switching is to minimize the relay time of the cells while still keeping the cell discard rate low.

The ATM switch handles in a brief description some basic functions. The cell forwarding from an inport to an outport is called *space switching*. Cells are temporarily stored in *buffers* when the switch can not forward the incoming cells instantly. Usually buffers are placed at points where many cell streams are multiplexed. *Queues* order the cells in the buffers, they are used for e.g. implementing priorities between different service categories.

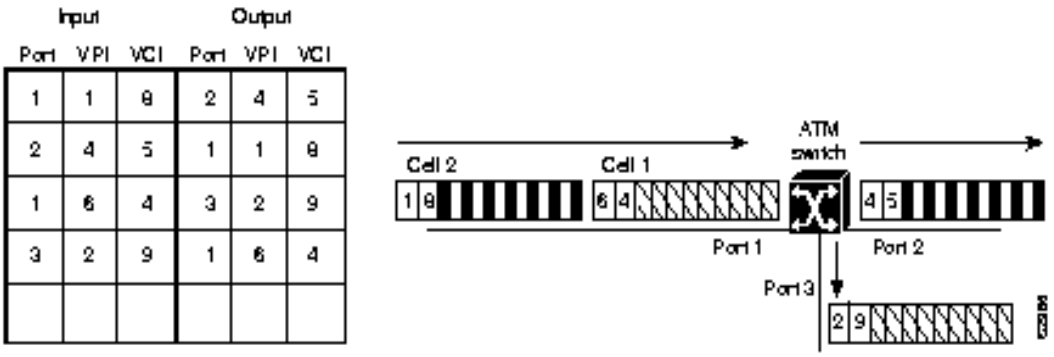


Figure 7. ATM switching

En example [15], two cells arrive at port 1 of the switch (figure 7). The switch table determines which output port each cell should be forwarded to and translates the VPI and VCI fields of outgoing cells. When the switch receives cell 1 on port 1 with the VPI

and VCI values of 6 and 4, the cell is relayed to port 3 with the VPI and VCI values translated to 2 and 9.

4.1.1 Conceptual model of an ATM switch

The architecture of ATM switches from different vendors may vary, but some basic building blocks can be identified in all switches. A basic conceptual model of an ATM switch is described below. The terminology may differ between different sources. In this report the following terminology are used, based on [16, 17, 18].

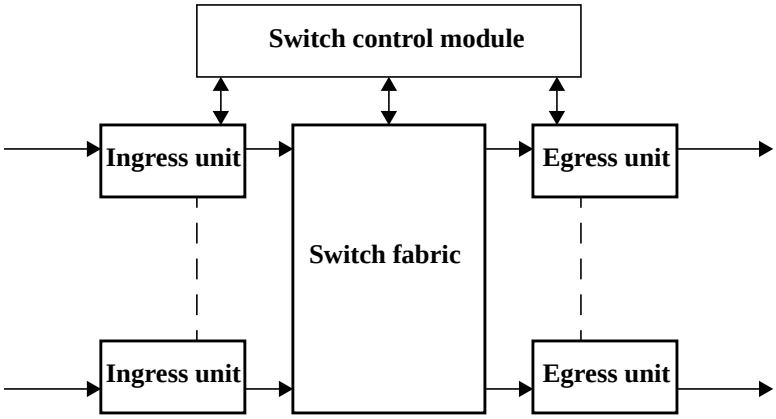


Figure 8. Conceptual model of an ATM switch

- *Ingress/Egress unit* (IU/EU)
- *Switch fabric* (SF)
- *Switch control module* (SCM)

The *ingress and egress units* are the interfaces between the ATM switch and the physical transmission links, sometimes the term exchange terminal (ET) is used as a common name for the ingress and egress units to the same link. The ingress and egress units should address the following functional areas:

- *Transmission and line termination* converts incoming optical signals to electrical signals and the reverse (ATM Transmission Convergence layer) and it performs synchronization and header error control (HEC) on the incoming cells.
- *ATM layer functions*, e.g. VPI/VCI look-up, buffering, traffic policing and congestion control are performed with support from the SCM.
- *Switch-core-interface*, e.g. adapting the cell format by adding and removing internal routing tags.
- *Signalling*, e.g. synchronize the timing of the switch with the network.

The *switch fabric* performs the space switching, it consists of a number of ingress and egress ports and a switch core. The cells are relayed from an ingress port to an egress port through the switch core. The switch fabric is built of many building blocks called

switch elements. In this report we use the terminology *switch fabric* for switch elements structured in a certain defined topology. The space switching is often done in two or more stages in the ATM switches of today.

The s*witch control module* (SCM) performs control, management and administration of the switch fabric. The SCM is composed of one or more *switch control processors* and functional software. The SCM handles fault, security and traffic management. Some important traffic control functions are:

- *Connection admission control (CAC) d*etermines if there are enough available resources to establish a new connection without violating the QoS of other connections. If there are enough resources available through the network to serve the the user's traffic needs, described by traffic and QoS parameters, a traffic contract is signed and a connection is established
- *Policing* discards or marks cells that exceeding the connection's traffic contract (PCR, SCR, MCR and MBS). The marked cells can be discarded later during periods of congestion. The police function is implemented differently for each service class.
- *Congestion control:* The switch has cell discard policies for controlling congestion in buffers. Service classes use different discard policies. Examples are selective cell discard (SCD), early packet discard (EPD) and partial packet discard (PPD).

4.1.2 ATM switch design issues

It is important that an ATM switch is scalable, this means it is easy to upgrade the switch to increased capacity demands caused by e.g. network grow or changed network topology. The traffic in the network is dynamic, so the traffic load in the switch can fluctuate in a broad range. With the above kept in mind, here are some important design issues for an ATM switch [17, 18]:

- *buffers*
- *organization of queues*
- *contention resolution*
- *support for performance measurement*

Buffers

The main reason for buffering cells in different stages of the switch is to temporarily store cells while waiting for busy resources. Generally, cells may be buffered at three stages, in the ingress unit, in the switch core and in the egress unit. *Ingress* buffers are important for resolution of contention in the switch core, which occur when the switch core relays more cells to the same egress port than it is possible to write in the egress buffer. By using buffers in the *switch core* cells can temporarily be stored to resolve the contention, then relay them in a later cell cycle. However, core buffers are limited in size why cells must be buffered in the ingress. *Egress* buffers are used when the cell arrival to an egress unit momentarily exceeds the rate it can send out on the physical link.

Organization of queues

By using queues the cells are ordered in the buffers. Dedicated queues for each service class are used to implement the priorities between service classes. A *queue-scheduler* is necessary at places where a selection of a cell must be done from a set of possible queues. Usually, a queue-scheduler is needed in two positions: between the ingress buffers and the switch core and between the egress buffers and the physical link interface. The organization of queues is a key factor for the performance of the switch. An example, a possible problem with using FIFO (first-in-first-out) queues for the ingress buffer is head-of-line blocking (HOL). If the first cell is blocked it will block all the following cells in the queue.

Contention resolution

It is necessary to have a mechanism for contention resolution, thus a way to control the rate of forwarding cells at some termination points, e.g. between the ingress buffers and the switch core and between the switch core and the egress buffers. Generally, there are two methods for contention resolution: the proactive and the reactive. In the *proactive* method the sender checks with the receiver before forwarding cells. In the *reactive* method the sender forwards cells until the receiver sending a signal indicating congestion. A queue-scheduler described in the previous section also includes a method for contention resolution.

Support for performance measurement

Measurements of QoS and NP parameters are based on counters in the hardware, some common counters are e.g. cell counters (transmitted, lost, errored etc.), and queue counters (e.g. length). These counters are used either direct or with some evaluation of many counters to represent measured values of QoS and NP parameters. An ATM switch should have hardware counters that give the possibility to measure important QoS and NP parameters.

4.2 The AXD301 switching system

A survey of the AXD301 from Ericsson Telecom AB exemplifies the architecture of an ATM switch [17, 18, 20].

4.2.1 The AXD301 switch architecture

The switch architecture is scalable from 10 Gbps to 2,5 Tbps. At the moment of writing this report the highest capacity of the AXD301 that can be delivered to customers is 40Gbps, so it is not the architecture that limits the capacity.

Switch circuit architecture

The basic building blocks in the AXD301 are the *switch port circuit* (SPC) and the *switch core circuit* (SCC). The *SPC* contains both an ingress and an egress port. Cells are received in the ingress port from a physical line interface, and then they are space switched to an egress port via the switch core. The SPC handles most of the ATM layer

functions and the switch core interface functions. The *SCC* has 32 input and 32 output ports. The switch fabric is built of SCCs interconnected in a certain topology. The possibility to use different numbers of SCCs in different topologies makes it possible to scale the switch core.

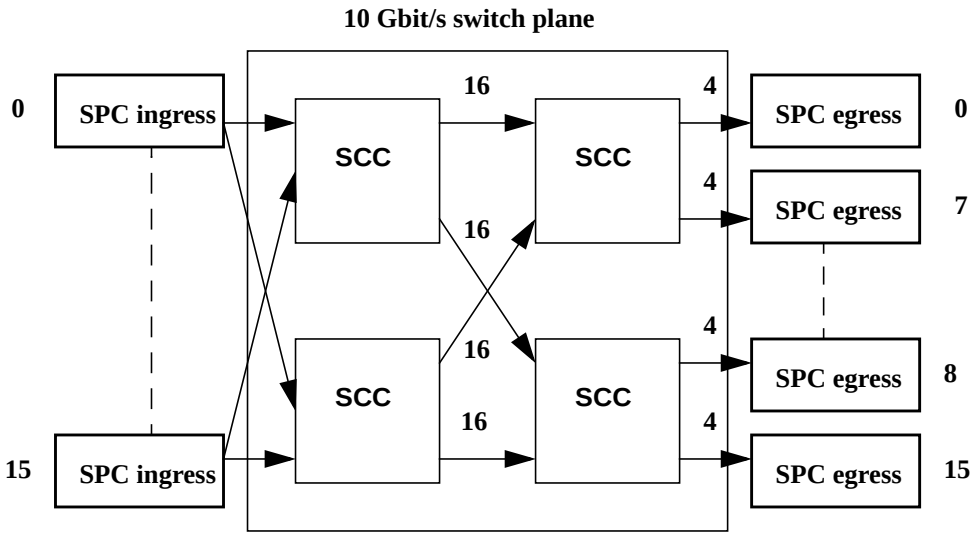


Figure 9. AXD301, 10Gbit/s switch plane

20 Gbps and 40 Gbps switch cores can be constructed from two or four 10 Gbps switch planes as the one depicted in figure 9. The 20 Gbps switch core has 32 SPC, where each SPC is connected to both 10 Gbps switch planes. And in the same way the 40 Gbps switch core has 64 SPCs connected to all four 10 Gbps switch planes.

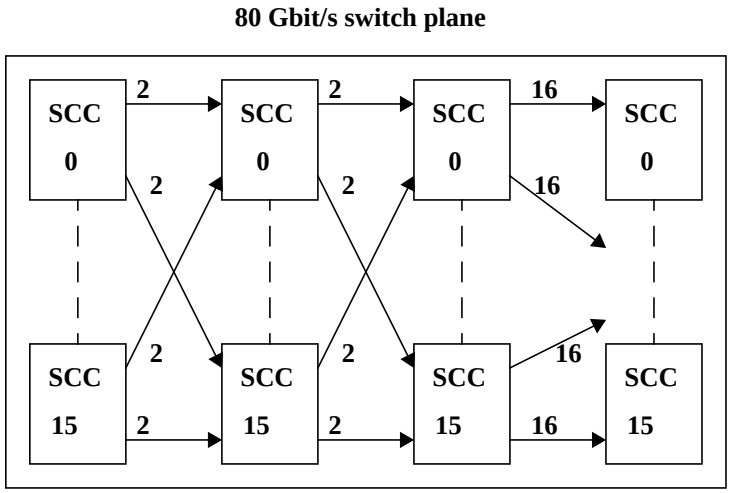


Figure 10. AXD301, 80Gbit/s switch plane

The 80 Gbit/s switch plane is built of 64 SCCs in four stages. It is possible to use one or two 80 Gbps switch planes in the switch fabric which makes 160 Gbps at most. With five stages of SCCs it is possible to build a switch core of 2,5 Tbps [17,18].

4.2.2 The AXD301 switch core

The switch core is built of two to five stages of interconnected SCCs. The switch core interface in the SPC appends an internal routing tag, which routes the cell to the correct egress port. A cell can take any path from the ingress port to a middle stage in the switch core. From the middle stage to the egress port the path is determined. The 10, 20 and 40 Gbps switch cores are built of two stages of SCCs. Cells are randomly distributed in the first stage, then in the second stage the cells are routed according to their internal routing tags. The random distribution in the first part minimizes the risk of collisions in the left part (the deterministic part), which occurs when to many cells are routed to the same SCC.

The switch core is *rearrangable non-blocking*. With any set of cells, it is possible to rearrange the paths between the ingress and egress ports so that all cells are switched correct. The random distribution has the purpose to rearrange the path between consecutive cells. The switch core handles *unicast* (point-to-point) traffic and *multicast* (point-to-multipoint) traffic differently.

Unicast

The switch core has no buffers for unicast traffic. A unicast cell enters and leaves the switch core within the same cell cycle, if no collisions occur. When too many cells contend for the same port, cells from lower priority levels are discarded first. Random selec-

tion is made between cells of equal priority levels. Priority information is contained in the cell header and there are three possible priority levels.

A positive acknowledgment (ACK) is signalled to the corresponding ingress port when the cell reaches the egress port. If the cell was discarded a negative acknowledgment (NACK) is returned to the ingress port. A discarded cell is retransmitted by the ingress port in some later cell cycle. The switch core operates at a speed 60% faster than the external links so it can handle increased load due to retransmission.

Multicast

The SCC contains buffers to handle multicast traffic. A multicast cell is stored in the buffers until all copies have been transmitted to the egress ports. The multicast identifier carried in the cell header are used to look-up a copy map. The copy-map specifies which egress ports the cell should be copied to. When the cell is copied, specific VPI and VCI values for the egress port is inserted in the cell header.

4.2.3 The AXD301 switch ports

Each SPC contains an ingress and an egress port. First the ingress ports receive cells from physical line interfaces then the cells are relayed to the egress ports via the switch core. Buffers ordered in queues are located both in the ingress and the egress part. The ingress buffers store cells due to e.g. bursty traffic and collisions in the switch core. The egress buffers store cells to synchronize them with the physical link rate

Organization of queues

The queues in the ingress ports are organized per connection. Queues that belong to the same service category are linked together in a second-order queue. This queue structure ensures fairness between connections and correct priorities between service categories. The queues in the egress port are organized per service class and physical link.

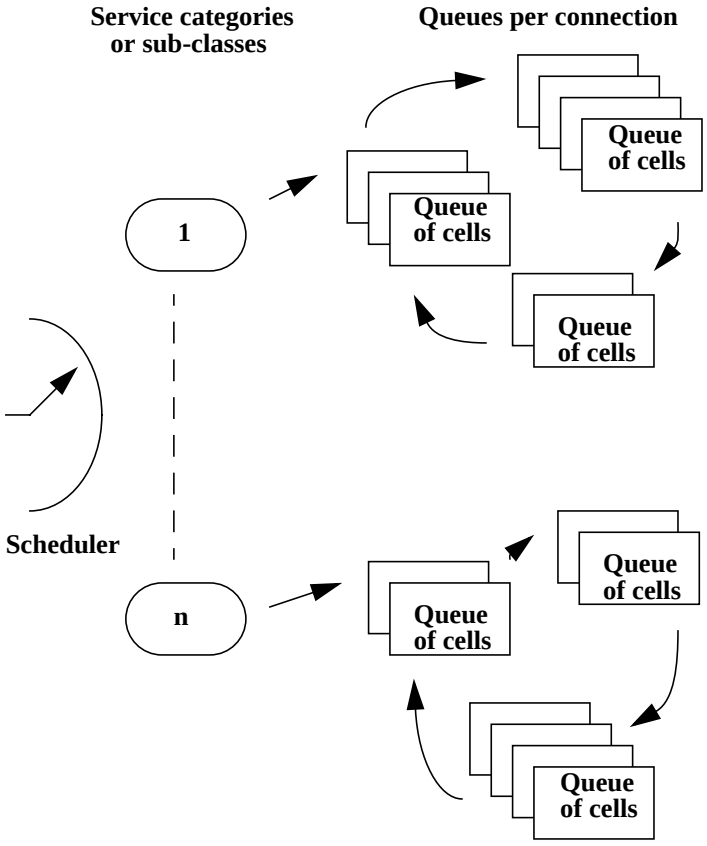


Figure 11. Organization of queues for the ingress ports

Queue scheduler

The *queue-scheduler* (scheduler) between the ingress buffers and the switch core selects a fixed number of cells to send into the switch core for each cell cycle. The scheduler chooses the service class with the highest priority and serves the connections in a round robin fashion. It continues to select service classes with decreasing priority levels until it has selected enough cells for the current cell cycle. It is possible to implement a large number of service classes with this approach. There are basically two different types of service classes: s*trict priority services classes* and *general purpose service* classes. The general purpose service classes are served only when there are no cells in the strict priority service classes. Further, weighting implements different priority levels within the strict priority and the general purpose service classes. Weighting specifies the number of cells to take every time the queue is served, it prevents a higher priority service class to totally starve a lower priority service class.

4.3 Management of the AXD301

Here follows a brief description of the possibilities to manage an AXD301 switching system, the information is solely extracted from the system description [17]. An AXD301 switching system can be managed in three ways:

- *SNMP* can be used for general network management and element management. The AXD 301 uses both IETF and ATM Forum standardized SNMP MIBs and Ericsson specified SNMP MIBs.
- *A standard web browser* is used to access the AXD301 element management system (AMS), which is a built in web server in the AXD301. The AMS can be accessed both remotely from a network management center (e.g. for performing switch configurations and monitoring) and locally by a computer directly connected (e.g. for hardware repair or upgrade work).
- *FTP* is used for sending bulk data, e.g. to get historical performance data and for input of software upgrades.

All management communication is carried inband over the ATM network.

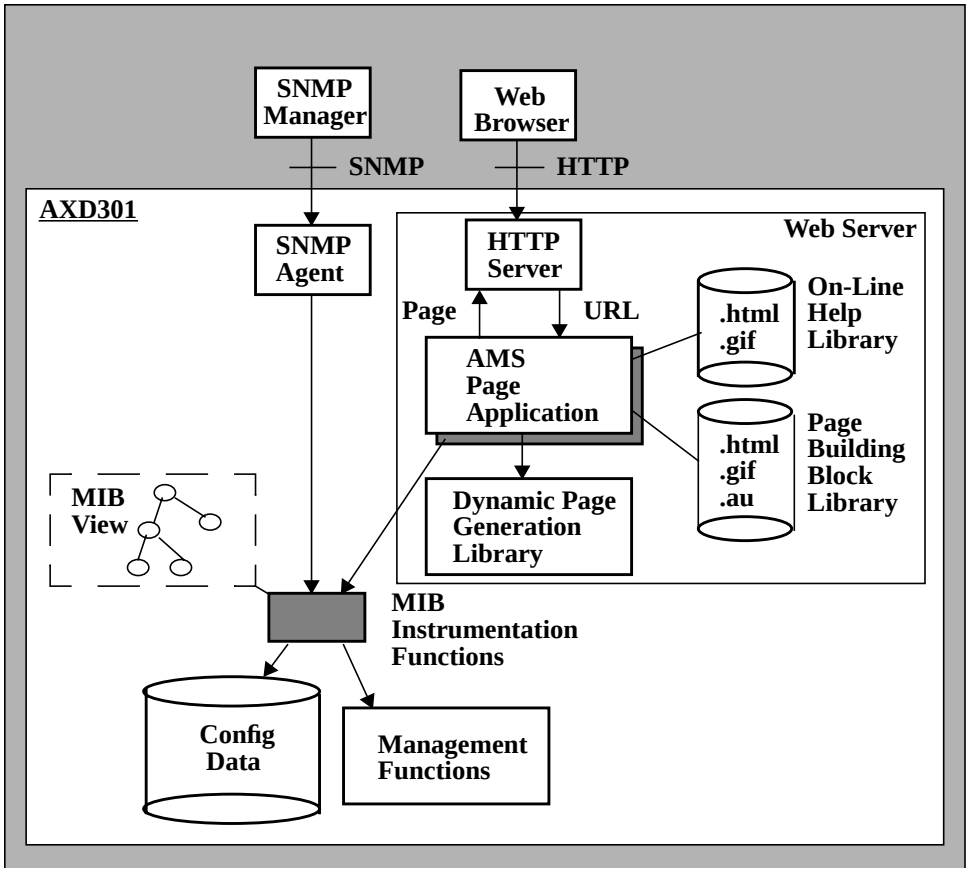


Figure 12. The AXD301 internal architecture for handling management communication.

4.3.1 The AXD301 Management system (AMS)

The graphical user interface to the AMS is based on HTML forms that the operator views in the web browser. There are input fields in the forms where the operator puts in values which initiates a management operation. The operator can get on-line documentation for the AXD301 through the AMS.

Management features in the AMS

Here is a brief description of the management features in the AMS:

- *Security management*: To start an AMS session the system requires a user id and a password, all actions are logged during the session.
- *Equipment management* views present hardware and the hardware configurations in the system and it performs some fault management, e.g. localize fault and view fault reports and diagnostics.
- *Alarm management* logs alarms internally and notifies the operator by forwarding alarms to the AMS alarm window and sending SNMP traps.
- *Performance management*: The operator can initiate a performance measurement by specifying a group of counters to be read at certain time intervals, during a specified period of time. The number of counters to take part in the measurement are limited to avoid heavy processing load. The performance data is collected in a file and is sent to the remote operator by FTP.
- *Charge management* records call statistics to be used for billing customers.
- *Software upgrade management* can upgrade a system software (with the exception from some low level software like the operating system) without stopping the normal system operation. FTP is used to send software from a remote location to the network node.
- *System management* handles functions related to the internal computer system, e.g. view SNMP configuration, view misc. information about the software (e.g. upgrade history, current software version), view the resource utilization of the control system (e.g. processing load, disk space, memory available etc.).

5.0 The Traffic performance monitoring (TPM) tool

A traffic performance monitoring (TPM) tool is a set of functions in the management system that gathers statistics in a selected point in the network, which e.g. could be a link or a virtual connection. Collected statistics are monitored for the operator in the graphical user interface (GUI) of the TPM tool.

A demonstrator of a GUI for a TPM tool has been implemented (referred as just TPM tool). The design of the GUI is focused on the organization of information and how the performance statistics are presented for the operator. The TPM tool is a fully stand-alone application that only has the purpose of demonstration. The TPM tool suggests new functions and new design of the GUI for future management systems.

A scenario with a service provider and customers who buy *virtual private networks* (VPNs) justifies the TPM tool. The service provider needs functions to monitor the bandwidth utilization in the network on different levels, e.g. the total utilization in a physical link as well as the utilization in a logical link in a customer's VPN. A TPM tool with such functions supports the service provider to reach higher bandwidth utilization in the network, that is of benefit to both the service provider and the customer.

5.1 Background

The service provider is the owner and operator of the network from which the customer (e.g. company, organization etc.) buys services and uses its network resources. Today many customers choose to lease lines from the service provider to interconnect geographically separated departments or offices. The private networks in the offices together with the leased lines build a VPN for the customer.

Poor utilization of network resources

Customers usually do not have a clear understanding of how much bandwidth they need, and to make sure they have enough bandwidth for their peak demands they usually take more bandwidth than necessary. A public service provider confirmed that many of its customers have weak knowledge of their bandwidth requirements, and also explained that its management system did not contain functions to measure the bandwidth utilization for a specific customer. When many customers lease permanent lines, the service provider often has poor utilization of network resources because the customers in average only use a small amount of their allocated bandwidth.

Selling bandwidth twice

If there is some way to reach a higher degree of bandwidth utilization in the network it would be of benefit to both the customers and the service provider. A service provider with a tool that can monitor the bandwidth utilization in a granular way has the possibility to sell some of the bandwidth twice, which means allocating more bandwidth than the real capacity of a link (over-allocation). It is possible because the customers do not fully utilize their allocated capacity at the same time and the effect of statistical multiplexing accumulates the unused allocated bandwidth. Probably, the customer can pay less for the same amount of bandwidth, when the service provider has the opportunity to

sell some of the bandwidth twice. Over-allocation introduces a risk of violating the customers' QoS, so the service provider must have methods or algorithms that estimate this risk and keep it under a certain limit. More about this in section 8.1. The TPM tool that has been implemented in this work would support the service provider to measure the bandwidth utilization in a granular way.

5.2 Requirements for the TPM tool

The TPM tool should monitor bandwidth statistics for the following purposes:

- Make it possible to take advantage of statistical multiplexing and reach higher bandwidth utilization by over-allocation.
- Control and verify services in logical network partitions (a set of virtual connections, e.g. a VPN).
- Reduce unnecessary bandwidth allocation in a logical network partition, by observing bandwidth statistics (e.g. a more efficient bandwidth allocation in a customer's VPN).

Requirements

Requirements for the GUI of the TPM tool:

- Possibility to monitor a real-time measurement of the bandwidth utilization in a physical or logical link (one or many virtual connections).
- Possibility to monitor bandwidth statistics of a physical or logical link.
- It should have a clear and intuitive presentation of bandwidth statistics, minimizing ambiguity.
- Possibility to concurrently show the physical network and a logical network partition (e.g. a customer's VPN) in separate views.
- It should be easy to navigate and see the relationship between different views.

5.3 The network scenario for the implementation of the TPM tool

A customer buys a *set of logical links* between nodes it wants to interconnect. Each logical link is realized with one or many permanent VPCs. The set of VPCs make up the customer's VPN on top of the service provider's public network. The customer decides how to use the VPCs. For example, each type of service can have a dedicated VPC into which the customer's traffic is multiplexed and transmitted over VCCs. The importance with this scenario (a customer's VPN) is the approach to set up a logical network partition with a set of permanent VPCs. Another possible scenario is a service provider that supplies a number of services, where each service has a dedicated logical network partition.

A customer is characterized by a number of users and a set of typical services. For example a hospital may be characterized by the number of doctors that could use services like e.g. sending x-ray pictures and patient journal. A simulation engine with a set of defined customer cases has been implemented to generate simulation data for the TPM tool, it is described in later chapters of this report.

A customer's VPN

The customer's private networks are interconnected with a set of VPCs. The customer adapts the equipment in the private networks with a couple of *private branch exchanges* (PBXs) and *service access multiplexors* (SAMs), so it can send traffic over the VPCs. The PBX is a switch that connects the customer's private network to the service provider's public network and offers connectivity inside the customer's private network. The SAM can multiplex multiple users and services into a single VPC.

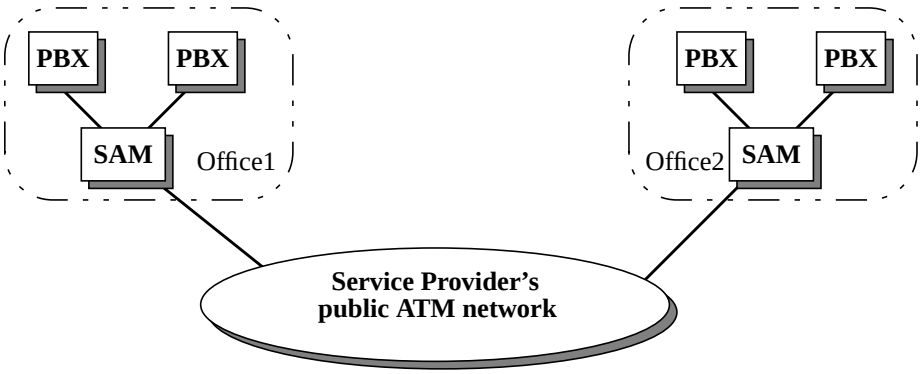


Figure 13. The customer equipment

The VPC between two customer nodes may pass through several switching nodes in the service provider's public network. The customer's perception of the connection is a single *logical connection* where all switching details in the network are hidden (figure 14).

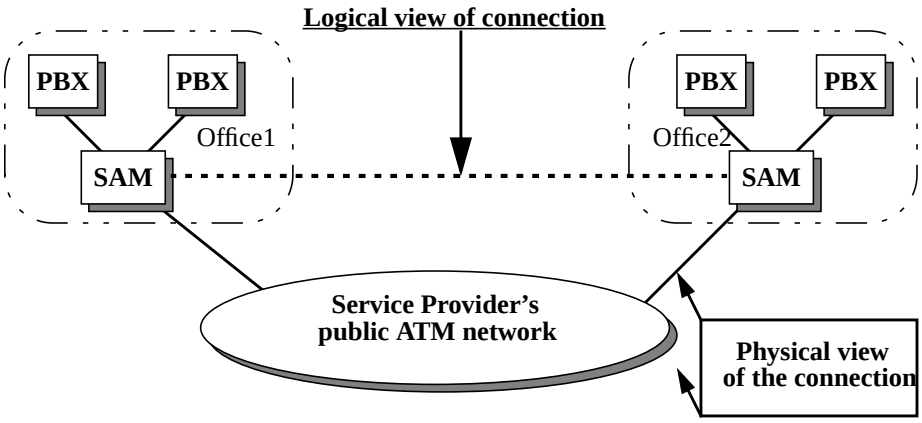


Figure 14. Logical network view

Organization of services

The VPCs should be organized per service rather than per user. One or many services with similar real-time demands becomes multiplexed into a common VPC, which is shared between many users. It is not a good approach to let a non real-time critical service (e.g. mail) compete with a real-time critical service (e.g. a bank transaction) for the same bandwidth. This would be the case if the VPCs was organized per user, where different services became multiplexed into a common VPC. A VPC for critical real-time services must be allocated to manage peak rate, while for file transfer services it can be sufficient to allocate less than peak rate.

6.0 Modelling of the TPM tool

The TPM tool consists of two main parts, the GUI and the simulation engine. The GUI is the operator's user interface and the simulation engine simulates the traffic in a fictitious network.

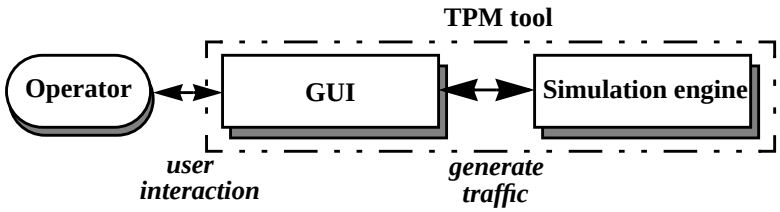


Figure 15. The TPM tool

The TPM tool is a stand alone application with the purpose of demonstration. Most effort is spent on parts in the GUI that have importance to the requirements stated in section 5.2. Important topics are the presentation of bandwidth statistics and how the GUI window is divided in different views.

6.1 The architecture of the TPM tool

The TPM tool is built as a client server application. The GUI is the client written in Java and the server written in Erlang is the part of the simulation engine that handles the communication with the Java client.

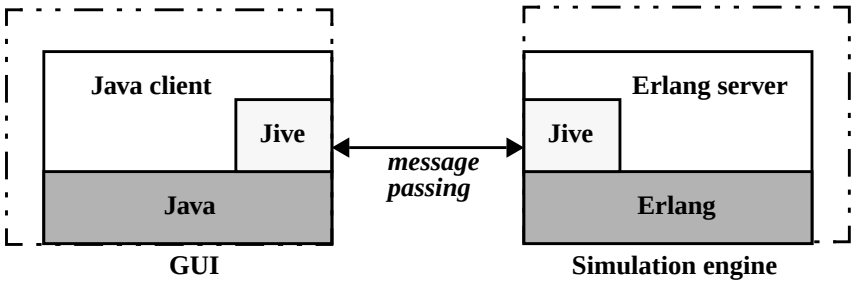


Figure 16. The architecture of the TPM tool.

GUI (Java client)

The GUI handles the interaction between the operator and the fictitious network (the simulation engine). It handles events like e.g. starting a real-time measurement of bandwidth utilization and presenting the bandwidth statistics for the operator. Java was chosen because it has good graphical support which is the main part of the GUI.

Simulation engine

The simulation engine generates (simulates) bandwidth statistics (alt. other performance data) to the GUI. The simulation engine is written in Erlang for two reasons: it is easy to make a robust server and much of the code in the AXD301 system is written in Erlang, so the choice was natural.

Client-server communication (using Jive)

The communication between the Erlang server (part of the simulation engine) and the Java client (GUI) uses the Erlang module *Jive* in the open telecom platform (OTP, an Erlang development and runtime platform [22]). A Jive server on each sides hides the underlaying socket communication, so the client and server can communicate through message passing. The conversion between Erlang data types and Java objects is done by wrapping each Erlang data type into a corresponding Java object on the Java side.

6.2 Modelling of the GUI

The GUI is divided into four sub windows called views (figure 17):

- The *Network view (Operator view)* gives a true (physical) picture of the network topology, including node and link details. This view can show the whole or a part of the network.
- The *Logical view (Customer view)* shows the allocated network resources that are dedicated to a logical network partition (e.g. a VPN). This view gives a logical picture of a set of allocated network resources, thus it hides irrelevant details for end-users like e.g. transport details.
- The *Statistics info view* presents the bandwidth statistics (e.g. min., max and mean values) collected from a bandwidth measurement and some basic information of the measurement.
- The *Performance monitor view* monitors a real-time measurement of bandwidth utilization in a diagram, the number of samples visualized in the diagram can be set by the operator.

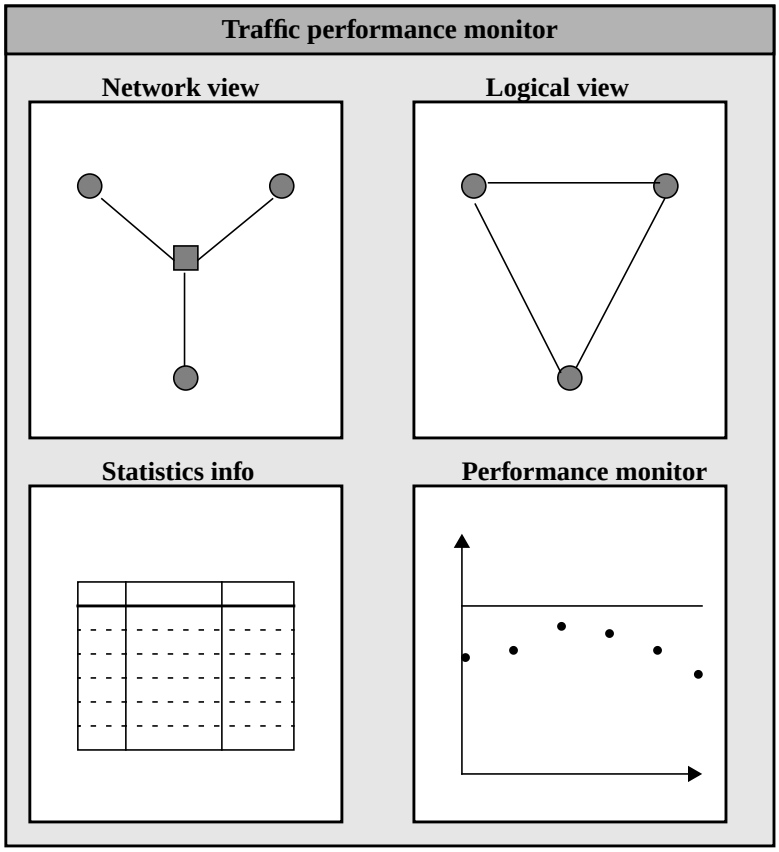


Figure 17. The organization of the GUI.

Operating the TPM tool

The network topology shown in the network view contains a set of logical network partitions (in this work customers' VPNs). A specific customer's VPN can be chosen from the network view to be visualized in the logical view. Link information (e.g. link capacity, a list with all connections, allocated bandwidth) are visualized when a physical or logical link is selected in the network or the logical view. It is possible to start a bandwidth measurement on the selected link where parameters like e.g. sample interval can be specified. The operator can monitor bandwidth samples (in the performance monitor view) and bandwidth statistics (in the statistics info view) for each connection within the link that is participating in a bandwidth measurement.

6.3 Modelling of the simulation engine

The simulation engine is simplified since its only task is to generate traffic (in this work just bandwidth samples) when a measurement is started. The simulation engine is general, so it is easy to define traffic with different characteristics.

Main features of the simulation engine

Each simulation is described by a *simulation case* that gives the characteristics to the generated traffic. A simulation case is described by a number of *users* and a *service profile* (a set of common services). A *counter* for each simulated connection keeps track of the traffic with a value that represents the current bandwidth. The simulation (a set of Erlang processes) causes the counter value to fluctuate in accordance with the defined simulation case. This is a simple and flexible simulation model that is sufficient for the simulation within this work. For example, in an office there are a number of workers. Each worker sometimes starts using some kind of service (e.g. sending a mail or a text document), which generates traffic on the office's network. It is easy to translate this real life example to the implementation of a simulation case.

Define a simulation case

A simulation case is defined by:

- *number of users*: Number of potentially concurrent users of a service. A user could be in either *idle* or *processing* state. In the processing state the user can in a random manner start using a service in accordance with the defined service profile, while in the idle state the user does not use any service.
- The *service profile* is a set of defined services with a corresponding probability to start. The traffic a service generates is described by two values, the *bandwidth* and the *duration time*.

When a user changes state from idle to processing it starts generating traffic in a random manner in accordance with the service profile. When the user goes from processing to idle it stops the traffic generation. This is depicted in figure 18. The approach with the user states simulates the *burstiness* of the traffic.

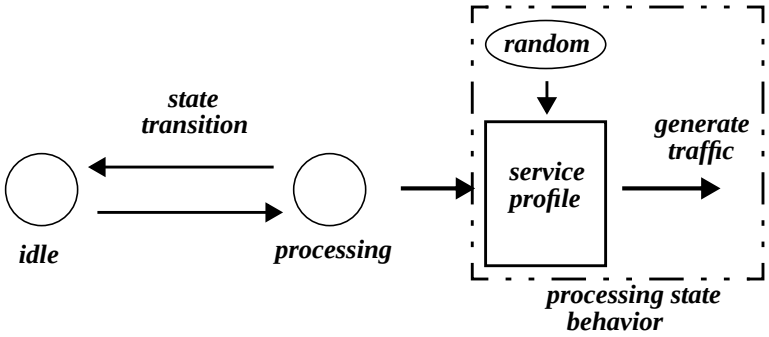


Figure 18. The user state transition

Service profile

The *service profile* is a set of traffic types defined by a bandwidth and a duration time (figure 19) where each traffic type is assigned a probability to occur. The traffic type is a description of how the traffic behavior of a service is modeled in the simulation. All traffic types in this simulation are modelled with fixed values for bandwidth and duration time. Simulating *unevenness* of the traffic can be achieved by chopping up a traffic type into a number of smaller traffic types (shorter duration times) with small variations of the bandwidth values. This is partly done in the implementation. The simulation can be made more indeterministic by randomizing the duration time each time a service is used (simulated), but that is not done in this work. The service profile (figure 20) can be defined with an arbitrary number of services and with the probability of each service arbitrarily defined. If the simulation needs to be more precise the number of traffic types can be extended.

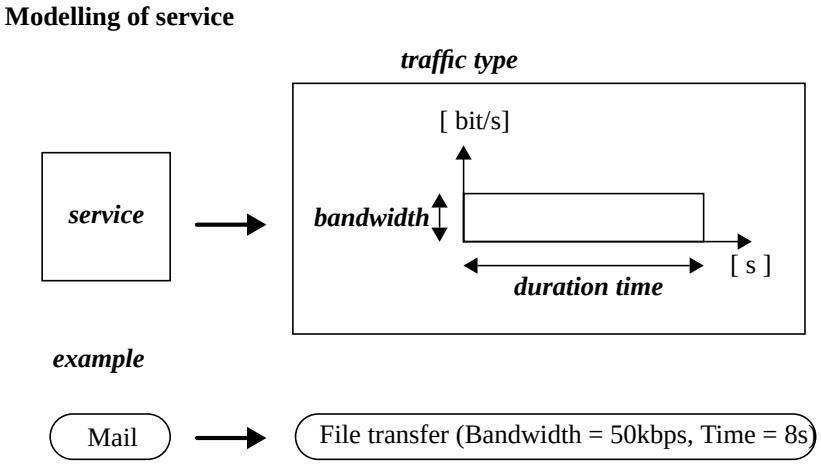


Figure 19. Service modelled into a traffic type

Service profile

service	*probability*	*traffic type*
mail	(5%)	(Bw = 50kbps, T=8s)
telephony	(10%)	(Bw=64kbps, T=120s)
.....		
video	(2%)	(Bw=2Mbps, T=300s)

Figure 20. Example of a service profile

Counter (measurement point)

The counter represents a measurement point of a connection (VPC or VCC). The counter supports three functions: *add*, *remove* and *return* current bandwidth (alt. other value).

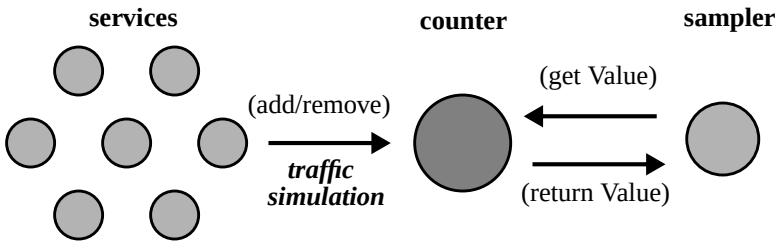


Figure 21. Modelling of a connection into a counter

When simulating a VPC that consists of a bundle of VCCs, the value of the VPC is the total sum of all VCC counters. All VCC counters must be read concurrently to give a correct value of the VPC. If it is some sliding in time between consecutive reading of the VCC counters the VPC value can be incorrect. It is not a problem in this work but has significance in a real system or in a more precise simulation. The simulation in this work is done on link (physical or logical) and VPC levels. Each link contains one or more VPCs, where each VPC has a counter. The link value is the total sum of all VPC counters.

6.4 Modelling a simulation case

A number of simulation cases are used in the simulation of the TPM tool. A simulation case is worked out from a sketched example with a customer and a corresponding service table. This is described in section 7.4. The *service table* contains examples of characteristic and common services used by the customer. From the service table a service profile is worked out. The *service profile* is used in the implementation of the simulation case.

Service types

Services are differentiated into three types:

- *File transfer*: Bulk data transfer, all data is sent at the same time. In the simulation the file transfer is modeled as even data transfer during a time interval, it is described with just the *data size* in the service table. Examples of file transfer are text documents, pictures and e-mail.
- *Real-time*: An application sends data continuously, where the data must reach the destination within a certain time. In the simulation the real-time application is modelled to have a fixed bandwidth during the application session and it is defined by the *bandwidth* and the *session time* in the service table.
- *Broadcast*: Sends data from one node to all reachable nodes. The data could be file transfer or a real-time application. In the simulation only real-time applications are used as examples of broadcast and it is defined by the bandwidth and the session time in the service table. A node using a broadcast service generates traf-

fic on all the outgoing links. Examples of broadcast applications are television and radio.

7.0 Implementation of the TPM tool

This chapter describes the main Java objects and Erlang processes and their relations. The intention is to illustrate the structure of important parts of the TPM tool without getting to deep into implementation (code) details.

General architecture

EApplet is a Java object inherited from the Applet class with the Jive interface implemented. An extended EApplet is the main object in the GUI, it is the top level window (applet) and handles the communication with the Erlang server, which contains the Jive interface on the Erlang side.

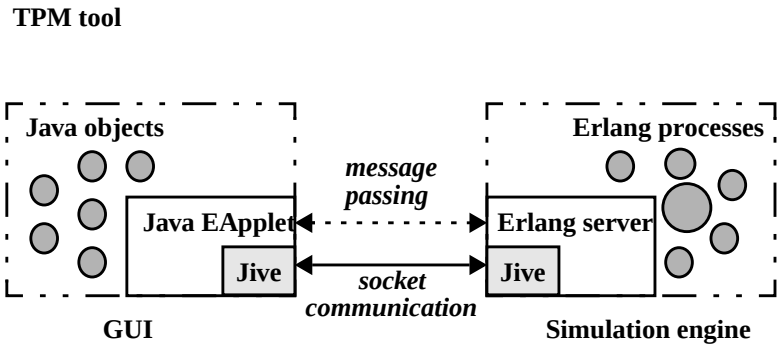


Figure 22. The architecture of the TPM tool.

7.1 Comprehensive solutions

Message agreement between the GUI and the Erlang server

The GUI and the Erlang server communicate through message passing with a set of predefined messages.The GUI can send messages to the Erlang server of type:

- The *start statistics* message is represented with the Erlang tuple {start_statistics, {StatType, MeasId, Interval, Connections}}. The argument tuple {StatType, MeasId, Interval, Connections} contains information about the simulation and the measurement (type of statistics, a unique identity of the measurement, sampling interval, a list of the connection objects that are participating in the simulation). This message is general, so it is possible to extend the TPM tool to simulate and measure other statistics than bandwidth, which is the only statistics considered in this work. Of course, some extensions must be done in the simulation engine to add new statistics in the TPM tool.
- The *stop statistics* message is represented with the Erlang tuple, {stop_statistics {MeasId}}. The argument {MeasId} identifies the measurement to stop.

The simulation engine can send messages to the GUI of type:

- The *sample* message is represented with the Erlang tuple {statistics, {MeasId, SampleValues}}. It sends back a sample of a measurement identified by the MeasId.

Simulation and measurement

The Erlang server in the simulation engine handles the communication with the GUI. When the Erlang server receives a start statistics message from the GUI, simulation and measurement processes (referred to with just *simulation process*) are started in the simulation engine. The Erlang server sends back samples from the simulation to the GUI, packed in sample messages.

The traffic simulation is simplified to only occur in links and connections that are involved in a measurement. This simulation approach is sufficient to demonstrate the TPM tool. It would not have been difficult to simulate the whole network but due to the limited time the simpler simulation approach was chosen. When the operator initiates a measurement the GUI sends a set of connections (possibly a link) and measurement information (e.g. type of statistics and sampling interval) aggregated in a *measure-type* object to the Erlang server which starts a simulation process.

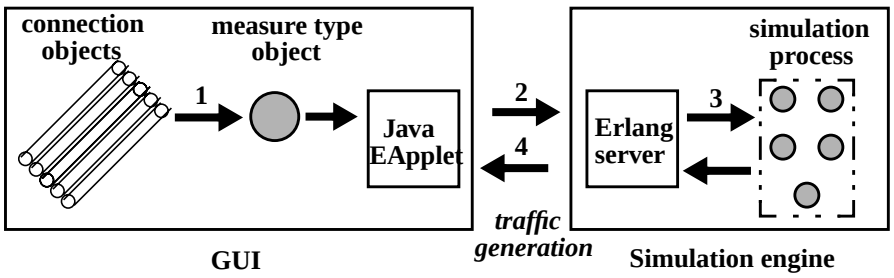


Figure 23. Start a simulation process.

7.2 Implementation of the GUI

GUI (client applet and top level window):

The main Java object is an extended EApplet (an Applet with the Jive interface) object that is the top-level window of the GUI. This Java object is also named *GUI*. The context should determine whether the term GUI relates to the Java object or the client part of the TPM tool. The window of the GUI (figure 24) is divided into four *views* (sub-windows):

- The *network view* contains a map that visualizes a physical network picture (topology) and a set of components (e.g. buttons, choices, information windows,

etc.) that supports functions and information on the level of network operation. This information level is typically aimed for an operator.

- The l*ogical view* contains a map that visualizes a logical picture of a set of allocated network resources (a logical network partition), thus it hides irrelevant details for an end-user like e.g. transport details. In this work a number of defined VPNs represent the logical network partitions. This view supports functions and information on a service level and this information level is interesting for both the operator and end-users.
- The *statistics info view* presents statistics of a link (physical or logical) measurement that can be started from the network or the logical view. It contains only information when a measurement is running otherwise it is blank.
- *The performance monitor view* presents real-time data in a diagram. It monitors the link (physical or logical) value along with one possible selected connection (e.g. a VP) within the link. The diagram is visualized only when a measurement is running.

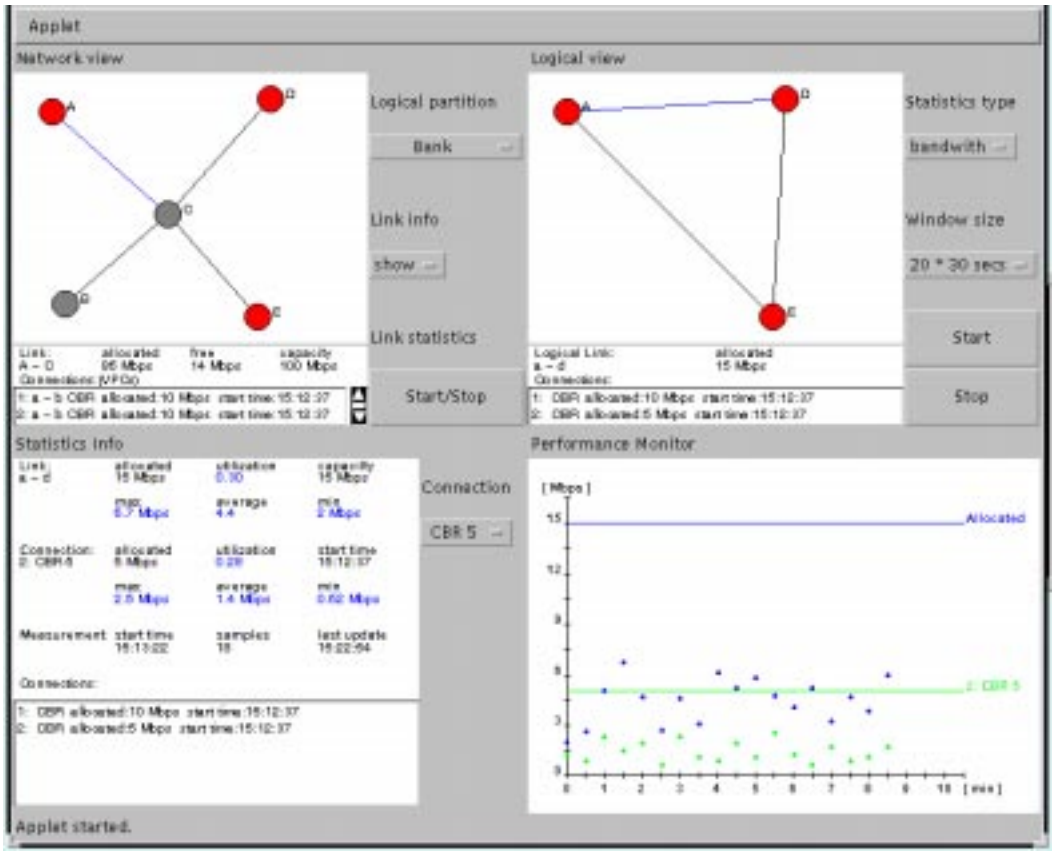


Figure 24. The GUI of the TPM tool

The TPM tool performs only measurement and monitoring of bandwidth statistics. A simple network topology consisting of five nodes connected with links is depicted in the

network view. Each node have a unique identity, here just a single letter. The central node (C) that interconnects the other nodes is a switch, while the other nodes (A,B,D,E) could be switches or just access equipment. A number of VPNs is set up in the network, each realized with a set of permanent VPCs between a set of access nodes. For each VPN a customer is specified (see section 7.4) and implemented as a simulation case in the simulation engine.

A certain VPN chosen by the operator (selected in the logical partition list) is visualized in the logical view. For each physical or logical link is it possible to: display link information and start a bandwidth measurement. *Link information* consists of: a unique identity, bandwidth capacity, allocated bandwidth and a list of set up connections within the link. The link information is visualized just bellow the network map. A *bandwidth measurement* monitors real-time values of bandwidth in the performance monitor and bandwidth statistics in the statistics info view. The bandwidth statistics is visualized for the link (physical or logical) and one possible selected connection (permanent VPC). The *bandwidth statistics* consist of:

- bandwidth capacity, allocated bandwidth, utilization (percentage use of allocated bandwidth). This group of values gives a measure of the *average utilization* of allocated bandwidth.
- min., max and average values (these values are based on the measurement samples). This group of values give a measure of *variations* in the usage of bandwidth.

The operator needs at least measures of the *average bandwidth utilization* and the *variations of bandwidth usage* to be able to make a decision (calculation) if it is possible to over-allocate the link. This work does not contain any deep analysis of what parameters (statistics) that would be the optimal for this kind of decision, but those bandwidth statistics presented in the statistics info view would at least be necessary. See section 8.3 for further discussion on this topic.

7.2.1 The Java objects

The characteristics and the relations between the Java objects in the GUI that implement important functions of the TPM tool are illustrated in object figures. Here follows a description of symbols and terms used in these figures. A *user interface object* is a window that usually contains some components that a user can interact with (e.g. buttons, text fields, canvas etc.). A *canvas object* is a basic graphical object that can be used to paint things on (e.g. when implementing a diagram or a map). An *aggregation object* simplifies the object administration by making many objects with natural relations into one single object. Arrows show the relations between objects and there are two types: *has-one* (just an ordinary arrow) or *has-many* (an arrow with a dot) relation.

The object relations for the *GUI object* that already has been described in the beginning of section 7.2 is depicted in figure 25. All four views in the GUI: the network view, the logical view, the statistics info view and the performance monitor view are implemented as separate objects.

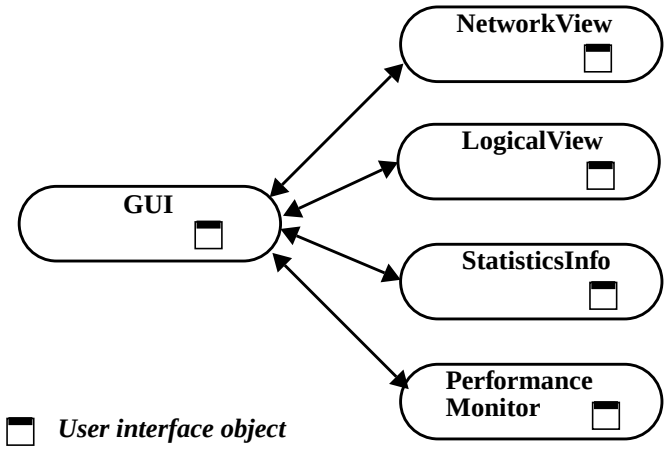


Figure 25. Object relations for the GUI object

The *network*- and *logical*-view objects (figure 26) are rather equal but they differ somewhat in the operation and information levels so they were made as separate object classes. The *network*- and *link-info*-canvas objects visualize the network map and link information. The *network-topology* object contains information of the network structure (nodes and links) in a single object.

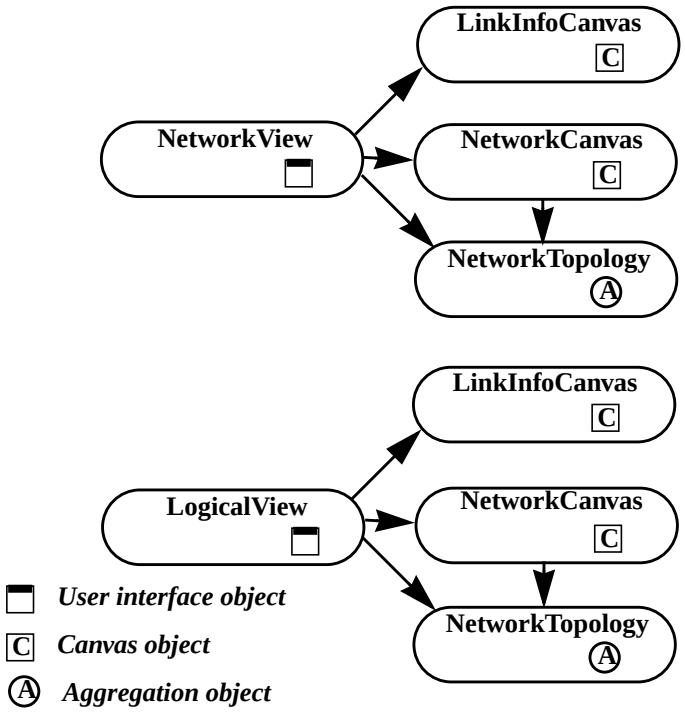


Figure 26. The NetworkView and LogicalView objects.

The *statistics-info* and *performance-monitor* objects are related to measurement activities (figure 27). The *measure-type* object contains information of how to perform the simulation and the measurement. The *statistics-window* object contains a data structure to store received measurement samples and information regarding the presentation of the samples.

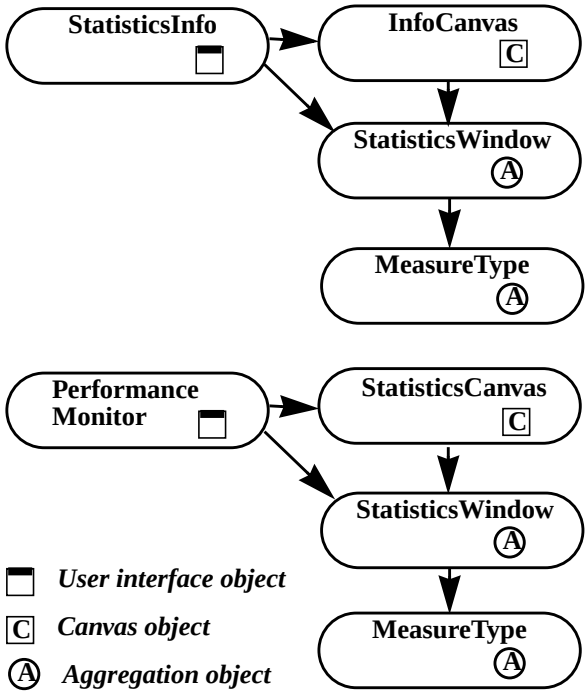


Figure 27. The StatisticsInfo and PerformanceMonitor objects.

Network topology

The *network-topology* object (figure 28) aggregates the network component (node and link) objects into a single object. It makes it easier to change the network picture in the network view and the logical view. The *link* object (figure 29) represents a physical or a logical link. It contains a bundle of connection objects and a set of methods (e.g. add and remove) which make it possible to change the number of connections dynamically. The *connection* object (figure 29) represents a VP or VC connection. It contains a *service-class* object (figure 29), which contains information like: service category (e.g. CBR), allocated bandwidth (PCR in the CBR case) and simulation case. The *node* object is basically used for graphic presentation.

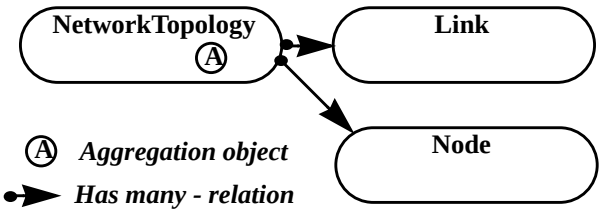


Figure 28. The NetworkTopology object.

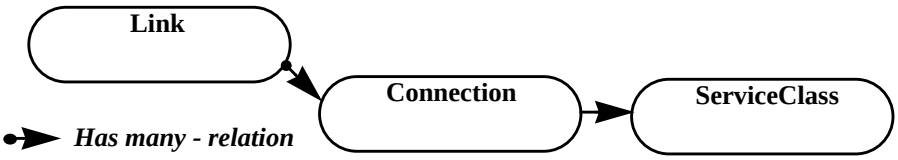


Figure 29. The Link object.

Starting a measurement

A measure-type and a statistics-window object is created when a measurement is started. The *measure-type* object (figure 30) contains a set of connection objects (possibly a link object) that participate in the simulation and information about the measurement, e.g. type of statistics and sample interval. This object is sent over to the Erlang server to start the simulation engine. The *statistics-window* object (figure 30) contains a set of fifo objects (fifo queues) to store measurement values and information regarding the monitoring of the values, e.g. *level* objects. The measure-type and statistics-window objects are unique for each started measurement and they are killed when the measurement is stopped.

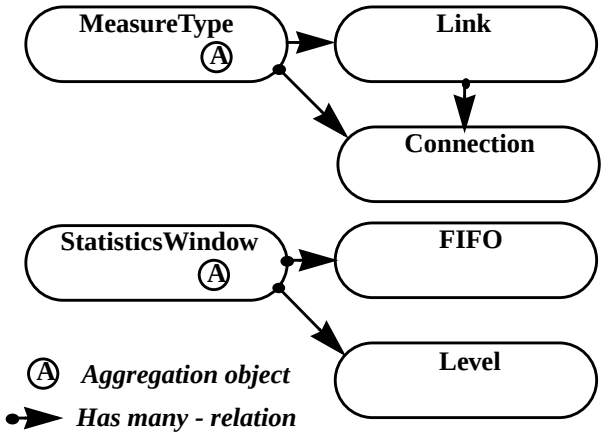


Figure 30. Starting a measurement creates a MeasureType and a Statistics Window object.

7.3 Implementation of the simulation engine

The simulation engine consists of the Erlang server and all Erlang processes created for a simulation (referred to as a simulation process). The Erlang server communicates with the GUI and is constantly alive, while the simulation process only is running during a measurement.

7.3.1 The main Erlang processes

The Erlang server can receive two types of messages from the GUI: *start statistics* and *stop statistics* (described in section 7.1). The start statistics message has an argument tuple {StatType, MeasId, Interval, Connections} (type of statistics, unique measurement id, sample interval, list of connections that are participating in the simulation) that initiates the simulation process. The simulation engine can run at most one simulation at a time. This approach was chosen due to limited time and that it is sufficient for the demonstration purpose of the TPM tool.

It would be easy to extend the simulation engine to handle multiple concurrently running simulations. Each simulation process is uniquely identified by the MeasId argument and can run independent of other concurrent running simulation processes. The Erlang server just needs to keep track of the identities of the running simulation processes. A similar extension needs to be done in the GUI part of the TPM tool with a measure-type and a statistics-window object mapped to each simulation process in the simulation engine.

Simulation process

The simulation process is initiated (figure 31) with the argument tuple {StatType, MeasId, Interval, Connections}. It creates a *simulation case* and a *counter process* for each connection in the argument Connections (list of connections). Each connection contains information that maps it against a simulation case. The simulation creates one *sample process* that reads all counter values concurrently (almost) with the interval defined in the argument Interval (figure 32). Then the aggregated sample is sent to the GUI.

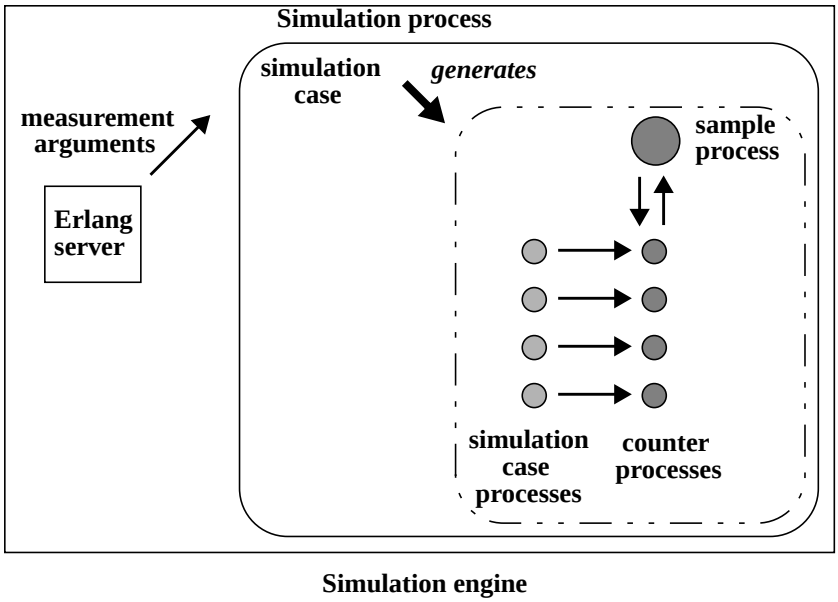


Figure 31. Process generation when starting a simulation process.

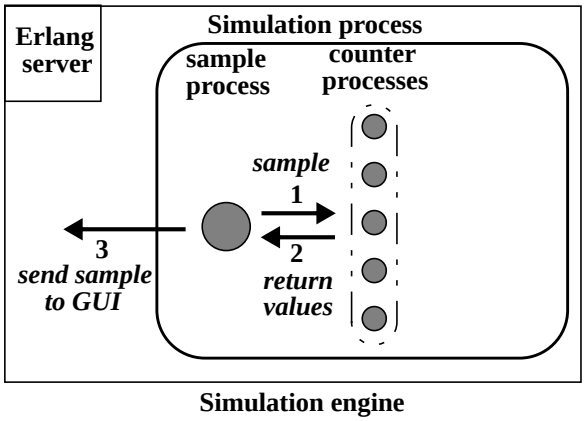


Figure 32. The procedure of measurement sampling.

Simulation case

A connection is described by the tuple {Category, CellRate, SimCase} that maps the connection to a *simulation case*. A simulation case (described in section 6.3) is defined by two arguments: *number of users* and a *service profile*. A user can be in an idle or a processing state. When the user is in the processing state a *service process* generates traffic in a random manner in accordance with the service profile (figure 33). *Traffic generation* is implemented by sending bandwidth values to the counter by message passing. A user in the idle state generates no traffic. The *user state transition* from idle to processing state has been implemented in two ways.

In the first solution, *the service process is alive in both user states* and the probabilies of state transitions are defined in the service profile. Thus, the idle state is simply implemented as a traffic type with zero bandwidth. This approach should preferably have a mechanism to vary the duration time of the idle state. A number of idle states (zero bandwidth) with different duration times were implemented in this work. Another way is to randomize the duration time each time the idle state is entered.

In the second solution, *the service process is alive only when the user is in the processing state*. A service process is created when a user enters the processing state and killed when the user leaves the processing state. A dedicated *state process* manages idle users to enter the processing state in accordance with an *arrival algorithm*. Here, uniform distribution in a defined interval is used as the arrival algorithm. The state transition from processing to idle is implemented by a defined probability in the service profile. Idle users only exist as a counter value in the state process. This second approach reduces the number of processes running during the simulation in comparison with the first solution.

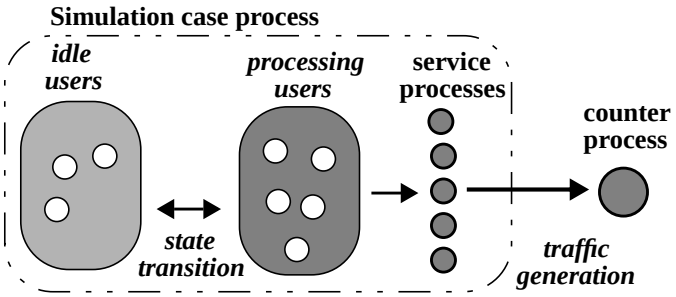


Figure 33. A simulation case process with a corresponding counter is created for each connection

Service profile

The service profile sets the characteristics of the generated traffic. It is implemented as a list with *traffic types* and a corresponding *probability.* The service process generates a uniformly distributed random number in a defined interval. The probability is implemented by dedicating a proportional part of the interval to each traffic type. For example, a service profile could look like this {[{Bw1, T1}, ..., {Bwn, Tn}], [N1, ..., Nn]}. The first list [{Bw1, T1}, ..., {Bwn, Tn}] contains the traffic types and the second list [N1, ..., Nn] divides the total interval into smaller intervals which give each traffic type a probability.

7.4 Definition of the simulation cases

A simulation case is defined for each set up VPN in the TPM tool. The simulation cases are chosen to represent a broad range of possible customers with different services. For each customer a number of *applications* (services) are listed in a *service table*. An application belongs to one of the following *service types* (described in 6.4): File transfer (F), Real-time (R) or Broadcast (B). The implemented service profiles are worked out from these service tables with some changes. For example, in a service table with a big difference in data size between services, a service with small data size and high frequency of occurrence is aggregated to a larger data size.

Hospital

In a hospital there are a number of doctors. A doctor can send patient journals between different divisions in the hospital, e.g. X-ray pictures, diagnosis and patient information. Sometimes a patient with minor diseases can get some advice from a doctor over the phone or by a video conversation.

application	service type
a few X - ray pictures	F(10MB)
patient journals	F(100kB)
remote treatment (by telephone or video conversation)	R(64kbps or 2Mbpx, 5min)
telephone	R(64kbps, 2 min)

Figure 34. Service table for a hospital.

Finance

A bank represents a customer from the finance sector. A bank is characterized by a lot of transactions generated by the bank customers, which also can perform bank services from a remote place with a web browser. The bank employees are a lot in contact with finance institutes and other banks.

application	service type
a transaction	F(1kB)
telephone	R(64kbps, 2min)
communication (telephone and data) with finance institutes and other banks	R(2Mbps, 30-60 min)

Figure 35. Service table for a bank.

Retail trade

A group of retailers for a common car trademark work as an example of the retail trade. One of the car dealers gets price listings or checks the stock for a certain car model from the general agency or from other car dealers. The car dealer occasionally generates a picture of a car specified by a customer and shows this picture to the customer. Every time the car company launches a new model or when something else important happens, the director for the car company has a video conference for all retailers.

application	service type
pricing lists and other documents	F(100kB)
tele cad “customer spec.”	F(1MB)
telephony	R(64kbps, 2min)
director talks	B(5Mbps, 10-60 min)

Figure 36. Service table for a car dealer

Manufacturer

A furniture manufacturer has a number of suppliers who deliver the basic material, e.g. wood and fabrics. Retailers or a wholesaler buy the complete furniture. Here are some examples of contacts: The manufacturer sends specifications and orders to the suppliers. The retailers order new furnitures. A retailer can be looking for a new furniture model, then the manufacturer arrange a video conference where a worked out proposition of a new furniture model is presented. A retailer shows a video for a customer of a furniture model they currently do not have in stock.

application	service type
orders and other documents	F(500kB)
design specification	F(2MB)
telephony	R(64kbps, 2min)
video conference	R(2Mbps, 30-60 min)
video “show model”	R(2Mbps, 5 min)

Figure 37. Service table for a furniture manufacturer.

Simulation examples

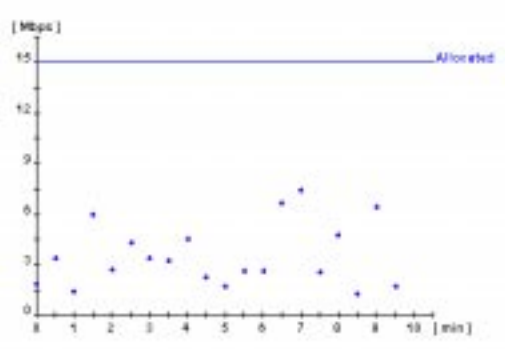


Figure 38. A simulation of the bank.

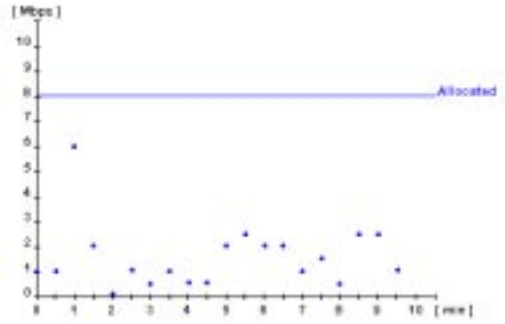


Figure 39. A simulation of the car dealer.

8.0 Evaluation and discussion of the TPM tool

8.1 Over-allocation based on the bandwidth statistics

The main purpose with the bandwidth statistics monitored in the TPM tool is to support decisions (calculations) to reach higher bandwidth utilization by *over-allocation*. The over-allocation approach introduces *a risk of violating the connections' QoS*. The operator must balance higher bandwidth utilization against an increased risk of violating the connections' QoS. The appropriate amount of over-allocation may vary from different cases depending on how critical the QoS are and the burstiness of the traffic. Bursty traffic with a large difference between peak and average bandwidth usually requires a greater allocated bandwidth to assure the QoS. In the case with a service provider, the violation of a customer's QoS would be connected with a fine for the service provider. There are mainly two ways to *over-allocate* links based on the bandwidth statistics:

- *Static over-allocation based on statistical calculations:* With this method the CAC functions have the perception of links with a bandwidth capacity greater than the real bandwidth capacity. In the AXD301 system there is an "over-booking" function that actually do this. The amount of over-allocation must be based on calculations of historical bandwidth statistics that estimates the risk of violating the QoS. The time periods for the historical bandwidth statistics used in such calculations may range from periods of about ten minutes to periods of about a number of hours or maybe even a number of days, it depends on how great the traffic variations are. A method like this must be rather cautious in the over-allocation, because it is very hard to get a good risk estimation based on just historical data.
- *Dynamic CAC*: In this approach the CAC functions make decisions based on the actual bandwidth utilization which can reduce the amount of unused allocated bandwidth from today's static CAC functions, where the decisions are based on just the allocated bandwidth. This dynamic CAC approach would require real-time bandwidth statistics like those suggested in the TPM tool. The implementation of dynamic CAC functions would be a complex task and how to modify the static CAC algorithm to a dynamic CAC algorithm is not covered in this report.

Functions like those in the TPM tool, where the operator can monitor real-time bandwidth statistics on different network (physical or logical) levels are a first and necessary start to reach higher bandwidth utilization. The second and the desirable step is to modify the static CAC functions to a more dynamic approach.

Further bandwidth statistics

The TPM tool monitors two groups of statistics: *average bandwidth utilization* and *variation of bandwidth usage* (see section 7.2). The parameters that describe the variation of bandwidth usage (min., max. and average values) are necessary for estimating the risk with over-allocation. These values are rather basic and statistics that describe the correlation would give a more exact risk estimation. This report does not contain how the parameters in the TPM tool can be analyzed to reach a more exact risk estimation. The goal with the bandwidth allocation is to allocate close to the average bandwidth utilization and still retain a small risk of violating the QoS.

8.2 Implementation of a TPM tool into a real system

It is a difference between the functionality that is desirable to have in a management system and the functionality that is possible and realistic to implement. This section summarizes issues to consider if similar functions like those in the TPM tool would be implemented into a real management system.

- *significance and validity of the implementation scenario*
- *collection of the statistics*

The areas are too extensive and complex to be investigated in depth within this report, this section just points out some possible problems.

Significance and validity of the implementation scenario

The implementation of the TPM tool is based on a scenario where the service provider is the owner and manager of the network and the customers buy services like VPNs (described in section 5.3). This scenario was chosen because *VPN* is a growing service among the service providers and the scenario also has relevance to other kinds of network operators than just service providers. Generally, an operator that has partitioned a physical network into smaller *logical networks* needs a tool to monitor the bandwidth utilization on different network levels (physical and logical). Here are some examples: a company that has partitioned its physical network into logical networks for each department and a network where each service has a set of dedicated (allocated) network resources. To put it in a context, an ATM network that is used as a multiservice network with a set of supported services, e.g. video, music and telephony where each service is allotted a bit of the network's resources. In all these described cases, the operator would be helped by functions similar to those in the TPM tool to monitor the bandwidth utilization in a logical link and the total bandwidth utilization in the physical link. The operator can use this information to *over-allocate* physical links, with a estimated risk of violating QoS (described in section 8.1). So a TPM tool would support the operator to reach a higher degree of bandwidth utilization. The TPM tool in this work only monitors real-time bandwidth statistics, even though historical bandwidth statistics are used in the risk estimation of a static over-allocation approach. Worth to mention is that historical data play an important rule for network planning and growth.

Collection of the statistics

The bandwidth samples in the TPM tool are monitored in a diagram in real-time. The samples are collected from the ATM node to the management system by a network management protocol, for example SNMP. A question that arises is how critical the real-time demand must be for the sample: is the sample too old when it reaches the management system? For real-time statistics, each bandwidth sample is transported with an exclusive SNMP message, while with historical statistics a set of bandwidth samples can be collected in the node before they are sent to the management system, transmitted by a single FTP transmission. It is much more efficient to transmit a set of bandwidth samples in bulk using FTP, than to use an exclusive SNMP message for each sample.

An *ATM node* must support *functions* for monitoring of bandwidth statistics. It must be able to sample bandwidth with a specified interval for both physical and logical link lev-

els. The processor/processors in the node have many other more important tasks (e.g. signalling and other ATM related tasks) to handle, so it might not be enough processor capacity available for monitoring real-time bandwidth statistics.

How much of the *analyzing* and *preparing work* of the bandwidth statistics should be performed in the node versus in the management center. More processing in the node reduces the traffic between the node and the management center, but the node may not have the required processing capacity. There is a trade-off between using node capacity or network capacity. It is usually preferable to do as much as possible of the statistics processing in the node. This issue is critical when considering a real implementation of the TPM tool.

9.0 Summary and Conclusions

The benefit

The TPM tool supports an operator to reach *higher bandwidth utilization* in a network by *over-allocation* in the links. It is mainly useful in networks with logical partitions where permanent connections are used, e.g. virtual private networks (VPNs). By over-allocation (allocate bandwidth twice) is it possible to make use of some of the already allocated but currently not utilized bandwidth. This introduces at the same time a *risk of violating the connections' QoS*. The TPM tool monitors *real-time bandwidth statistics* that support: decisions for over-allocation with a corresponding risk estimation, verifications of the users' traffic and monitoring of network performance (just bandwidth). This report mentions two ways to perform over-allocation: *static over-allocation based on statistical calculations* and *dynamic CAC functions.* It does not contain any evaluations and estimations of what would be an appropriate over-allocation. Functions similar to those in the TPM tool would be valuable for a service provider that sells VPNs.

The TPM tool

The GUI of the TPM tool is divided into *a physical and a logical network view,* which is important to be able to monitor bandwidth statistics on different network levels. The network view visualizes the physical network level while the logical view visualizes a set of logical organized network resources (a logical network partition). The logical view hides the irrelevant details to end-users' perspective, e.g. transport details. The TPM tool can monitor bandwidth statistics on both link- (physical and logical) and connection (VP and VC) levels, which is necessary to be able to distinguish the statistics between e.g. different customers or services. The bandwidth statistics measure *average bandwidth utilization* and some values that indicates the *variation of bandwidth usage*. These parameters are rather simple but are fundamental for calculations of more complex bandwidth statistics. This report does not contain any investigation of what bandwidth statistics that would be optimal to monitor.

Implementation into a real system

The implementation of functions similar to those in the TPM tool into a real management system requires some considerations concerning capacity issues and other possible problems. They are briefly mentioned but not evaluated in this report. Some topics to consider are: how much traffic the collection of statistics generates and where most of the statistics processing should be performed (in the node or in the management center). These aspects were not considered in the design of the TPM tool.

Future work

Because the TPM tool is a stand-alone application made just for demonstration purpose, suggestions for future work becomes rather restricted. Here are some possible ways to continue:

- Investigate what bandwidth statistics that would be optimal to base decisions for over-allocation on, e.g. covariance etc., and describe how these statistics are evaluated.

- Investigate and suggest an algorithm suitable for static over-allocation and describe how the parameters used in this algorithm are evaluated.
- Look at how to modify the static CAC functions to dynamic CAC functions, or at least suggest some small adjustments that give the CAC a more dynamic approach.

References

[1] Fore System, Inc, The Case for the Intelligent Infrastructure, August 19, 1998, http://www.fore.com/products/wp/iipaper/iipaper.html

[2] Case J, Fedor M, Schoffstall M, Davin J, “The Simple Network Management Protocol”, RFC 1157, May 1990

[3] ITU-T Recommendation X.711 (3/91), Common management information protocol specification for CCITT applications

[4] Fore System,Inc., ISO NMS Framework - FCAPS Overview, October 20, 1998,

http://www.fore.com/products/4vw/fcaps.htm

[5] ITU-T Recommendation I.350 (3/1993), General aspects of quality of service and network performance in digital networks, including ISDNs.

[6] Fore System, Inc., Bennett G, ATM Quality of Service part I, April 9, 1998, http://academy.fore.com/tutorials/html_files/qos1.html

[7] ITU-T Recommendation I.356 (11/1993), B-ISDN ATM layer cell transfer performance.

8] Waldbusser S, “Remote Network Monitoring Management Information Base”, RFC 1757, February 1995

[9] Fore System Inc., “ATM Management Interface (AMI) Manual”, MANU0021-05-Rev, A-8/17/98, http://www.fore.com/products/manuals.htm

[10] Fore System Inc., “ForeView, Network Management Software”, http://www.fore.com/products/4vw/index.html

[11] Fore System Inc., “Fore System, Products That Build Networks of Steel”, http://www.fore.com/products/psindex.html

[12] Fore System Inc., “ForeView Network Management User’s Manual”, MANU0363-01 - September, 1998, http://www.fore.com/products/manuals.htm

[13] Siemens Corp./Newbridge Networks Corp, Data sheets for the MainStreetExpress product line, http://www.newbridge.com/products/index.html

[14] Newbridge Networks Corp., 36150 MainStreet Maintenance , technical update, Release 2.3,, NNP 95-1318-01-00-B, http://prodweb.newbridge.com:80/updates/36150/index.html

[15] Cisco System Inc., Internetworking technology overview, Asynchronous Transfer Mode, 1998, http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/55755.htm#42544

[16] de Prycker M., Asynchronous Transfer Mode Solution for Broadband ISDN, 1995, ISBN 0-13-342171-6

[17] Ericsson Telecom AB, “AXD301 ATM Switching System, System Description”, 1998-08-02, 1551-AXD 301 01/1 Uen, Rev J

[18] Buhrgard M., Westin T., Wicklund G., “A scalable ATM switching solution”, Ericsson Review No 1, 1998, ISSN 0014-0171

[19] ATM Forum, Traffic Management Specification version 4.0, af-tm-0056.000, April, 1996

[20] Axell J., Hellstrand F., “ATM traffic management and resource optimization”, Ericsson Review No 1, 1998, ISSN 0014-0171

[21] ATM Forum, Remote Monitoring MIB Extension for ATM Networks, AF-NM-TEST-0080.000, February 1997

[22] Torstendahl S., “ Open telecom platform”, Ericsson Review No 1, 1997

Acronyms

AAL	ATM Adaptation Layer
ABR	Available Bit Rate
ACK	Acknowledgment
AMI	ATM Management Interface
AMS	AXD301 Management System
API	Application Programming Interface
ATM	Asynchronous Transfer Mode
CAC	Connection Admission Control
CBR	Constant Bit Rate
CDV	Cell Delay Variation
CER	Cell Error Ratio
CLR	Cell Loss Ratio
CP	Control Processor
CMIP	Common Management Information Protocol
CSM	Customer Service Management
CTD	Cell Transfer Delay
CMR	Cell Misinsertion Rate
DBMS	Data Base Management System
DP	Device Processor
EPD	Early Packet Discard
ERTS	Erlang Run-Time system
ET	Endline Terminal
FIFO	First In First Out
FTP	File Transfer Protocol
GUI	Graphical User Interface
HEC	Header Error Control
HOL	Head of Line blocking
ISO	International Organization for Standardization
LAN	Local Area Network
NIAB	Network Information Access Modules
NMC	Network Management Center
NMS	Network Management System
NP	Network Performance
MBS	Maximum Burst Size

MCR	Minimum Cell Rate
MIB	Management Information Base
MNSC	Multi Network Service Controller
MP	Measurement Point
NACK	Negative Acknowledgment
OTP	Open Telecom Platform
PBX	Private Branch Exchange
PCR	Peak Cell Rate
PPD	Partial Packet Discard
QoS	Quality of Service
RMON	Remote Monitoring Management Information Base
SAM	Service Access Multiplexer
SCC	Switch Core Circuit
SCM	Switch Core Module
SCP	Switch Control Processor
SCR	Sustainable Cell Rate
SCS	Switch Control Software
SECBR	Severely Errored Cell Block Ratio
SLA	Service-Level Agreement
SNMP	Simple Network Management Protocol
SPC	Switch Port Circuit
SYSLS	Systems Architecture Support Libraries
TPM	Traffic Performance Monitoring
UBR	Unspecified Bit Rate
VC	Virtual Channel
VCC	Virtual Channel Connection
VCI	Virtual Channel Identifier
VP	Virtual Path
VPC	Virtual Path Connection
VPI	Virtual Path Identifier
VPN	Virtual Private Network
VPT	Virtual Path Termination

Complementary reading

Litteratures

Stallings W.,Data and Computer Communications, 5th ed, 1997,ISBN 0-13-571274-2

Boisseau M., Demange M., Munier J-M.,ATM Technology An Introduction, 1996, ISBN 1-85032-304-6

Clark Martin P., ATM Networks Principles and Use, 1996, ISBN 0 471 96701 7

Pitts J M., Introduction to ATM Design and Performance, 1996, ISBN 0 471 963402

de Vries R. J. F., Switch Architectures for the Asynchronous Transfer Mode, 1992, ISBN 90-72125-33-9

de Prycker M., Asynchronous Transfer Mode Solution for Broadband ISDN, 1995, ISBN 0-13-342171-6

Reports

Norrgren M., Estimerig av kvalitetsparametrar i ATM-nät, May 1998, Masters's thesis Department of Teleinformatics, KTH

Stranden P., Performance Analysis of an ATM Switching System, October 1998, Master's Thesis Computer Science, Uppsala University

Documents(manuals, tutorials and white papers)

ATM technology

Ericsson Review, The Telecommunications Technology Journal, No 1, 1998, ISSN 0014-0171

Brochure: Ericsson Telecom AB, AXD301 High-performance ATM switching system,1998

Cisco System Inc., Internetworking technology overview, Asynchronous Transfer Mode, 1998, http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/55755.htm#42544

Management system

Brochures: Ericsson Telecom AB, Datacom Networks & IP Services, Multiservice Management Suite, updated 980911, http://www.ebc.ericsson.se/bn_dnip/productsmulti_service_management_suite/announcement/index.htm

Cisco Systems, Inc., Cisco Service Management System, September 29, 1998, http://www.cisco.com/warp/public/146/september98/25.html

Cisco Systems, Inc., Cisco Service Management System, September 29, 1998, http://www.cisco.com/warp/public/146/september98/25.html

Fore Systems, Inc.,ForeView, Network Management Software, August 19, 1998, http://www.fore.com/products/4vw/index.html

Fore System, Inc, The Case for the Intelligent Infrastructure, August 19, 1998,http://www.fore.com/products/wp/iipaper/iipaper.html (Infrastructure for the future)

Cisco System, Inc., Network Management Basics, 1995, http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/55018.htm

Fore System,Inc., ISO NMS Framework - FCAPS Overview, October 20, 1998, http://www.fore.com/products/4vw/fcaps.htm

ATM Forum, “M4 Network-View Interface Requirements, and Logical MIB”, AF-NM-0058.000, 9703

ATM Forum, “M4 Interface Requirements and Logical MIB: ATM Network Element View”, STR-NM-M4NE-REQ-02.00, july 98

Performance management

ITU-T Recommendation I.350 (3/1993) , General aspects of quality of service and network performance in digital networks, including ISDNs.

ITU-T Recommendation I.356 (11/1993), B-ISDN ATM layer cell transfer performance.

ATM Forum, Traffic Management Specification version 4.0, af-tm-0056.000, April, 1996

Fore System, Inc., Bennett G, ATM Quality of Service part I, April 9, 1998, http://academy.fore.com/tutorials/html_files/qos1.html

Fore System, Inc., Bennett G, ATM Quality of Service part III, July 1, 1998, http://academy.fore.com/tutorials/html_files/qos3.html

Performance monitoring

Fore System, Inc., Establishment of Multi-Vendor ATM Monitoring Group, July 5, 1995, http://www.fore.com/press/archive/1995/PR57_05.html

Waldbusser S, “Remote Network Monitoring Management Information Base”, RFC 1757, February 1995

Case J, Fedor M, Schoffstall M, Davin J, “The Simple Network Management Protocol”, RFC 1157, May 1990

ITU-T Recommendation X.711 (3/91), Common management information protocol specification for CCITT applications

ATM Forum, Remote Monitoring MIB Extension for ATM Networks, AF-NM-TEST-0080.000, February 1997

Fore System, Inc., “RMON FAQs”, 98-10-30, http://www.fore.com/products/4vw/fvr-monfaq.htm

NetScout System inc., “The Integration of RMON and SNMP Technologies for Managing Enterprise Networks”, white paper, 1997, CC-0014-97/B, http://www.fore.com/products/4vw/netscout.htm

Cisco System, Inc., “Simple Network Management Protocol”, http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/55029.htm

ATM Forum, “SNMP M4 Network Element View MIB, AF-NM-0095.001

ATM Forum, “Remote Monitoring MIB Extensions for ATM Networks”, AF-NM-TEST-0080.000, February 97

Siemens/Newbridge

Technical Updates for MainStreetXpress 46020 Network Manager, 46020 MainStreet Network Manager: http://prodweb.newbridge.com:80/updates/46020/index.html

Newbridge Networks Corp., 46020 MainStreet Network Management, 30.2 Network Performance Monitor, technical update, August, 1998, NNP 95-1829-01-00-L

Newbridge Networks Corp., 46020 MainStreet Network Management, Appendix A, Statistics Details, technical update, August, 1998, NNP 95-1829-01-00-L

Newbridge Networks Corp., MainStreetExpress 46020 MainStreet Performance Management, 63.9 Statistics Details, technical update, July, 1998, NNP 95-2199-01-00-K

Newbridge Networks Corp., 36150 MainStreet Maintenance , technical update, Release 2.3,, NNP 95-1318-01-00-B, http://prodweb.newbridge.com:80/updates/36150/index.html

Siemens Corp./Newbridge Networks Corp, Data sheets for the MainStreetExpress product line, http://www.newbridge.com/products/index.html

Fore

Fore System Inc., “ForeView Network Management User’s Manual”, MANU0363-01 - September, 1998, http://www.fore.com/products/manuals.htm

Fore System Inc., “ATM Management Interface (AMI) Manual”, MANU0021-05-Rev, A-8/17/98, http://www.fore.com/products/manuals.htm

Fore System Inc., “ForeView, Network Management Software”, http://www.fore.com/products/4vw/index.html

Fore System Inc., “Fore System, Products That Build Networks of Steel”, http://www.fore.com/products/psindex.html

Atm switches

Ericsson Telecom AB, “AXD301 ATM Switching System, System Description”, 1998-08-02, 1551-AXD 301 01/1 Uen, Rev J

Buhrgard M., Westin T., Wicklund G., “A scaleable ATM switching solution”, Ericsson Review No 1, 1998, ISSN 0014-0171

Axell J., Hellstrand F., “ATM traffic management and resource optimization”, Ericsson Review No 1, 1998, ISSN 0014-0171

Datacom Networks & IP Services at Ericsson Telecom AB, Product Portfolio - AXD301 Switching System, 9/17/98, http://bn-dnip.ericsson.se/bn_dnip/products/axd_301.htm

Software

Torstendahl S., “ Open telecom platform”, Ericsson Review No 1, 1997

Erlang 4.7.3/OTP R4B release, 980812, http://otp.ericsson.se:8000/product/info/otp_unix_r4b/doc/