Web Based ATM PVC Management

Bram van der Waaij, Ron Sprenkels, Bert-Jan van Beijnum, Aiko Pras
Centre for Telematics and Information Technology (CTIT), University of Twente
PO Box 217, 7500 AE Enschede, The Netherlands
waaij@cs.utwente.nl - tel.: +31-53-4894663

Abstract

This paper discusses the design of a public domain web based ATM PVC Management tool for the Dutch SURFnet research ATM network. The aim of this tool is to assists in the creation and deletion of PVCs through local and remote ATM network domains. The tool includes security mechanisms to restrict the access to these ATM domains, and provides a high level graphical user interface to hide the differences between the ATM switches from various vendors.

keywords: ATM, PVC, cross-connect, network management, web based, Java, SURFnet, multi vendor

1 Introduction

The infrastructure that provides ATM PVC connectivity between universities in the Netherlands is shown in Figure 1. The backbone of the network consists of three Lucent switches, which belongs to the dutch PTT. The access network is also owned by the PTT, and consists of a number of GDC switches. Each of the access switches connects a single university. The ATM access and backbone infrastructure is rented by SURFnet bv., who acts as intermediate between the dutch PTT and the various universities. Although SURFnet does not own the physical ATM infrastructure, SURFnet is seen by universities as an ATM service provider. This means, for example, that universities direct their requests to establish or remove PVCs to SURFnet, and to not the PTT.

Setting up an end-to-end PVC between different universities requires the configuration of all the intermediate switches on the path. Such configuration involves the creation of *cross-connects* in every switch. Because the switches come from multiple vendors, each with a proprietary management interface, it is a cumbersome task to set up such a PVC. If, for example, a member of the University of Twente wants to set up a PVC with someone at the University of Utrecht, at least six ATM switches must be configured: a CISCO LS1010 in Twente, two GDC switches in the access network, two Lucent switches in the backbone network and at least one ATM switch in Utrecht. It is clear that setting up a PVC is a complex task and that tools to make this task easier are needed.

This paper describes the development of a public domain ATM switch management tool for SURFnet bv. [3][4]. The tool has a high level user interface that allows managers with limited ATM background to manage cross-connects within ATM switches from different vendors in a uniform and secure manner. The tool allows management of ATM switches in the local domain, as well as switches in other domains. It is therefore possible that managers within the University of Twente create cross-connects in for example the ATM switches of the University of Utrecht. To restrict and control the access of user from other domains, the tool includes authentication, encryption and access control mechanisms.

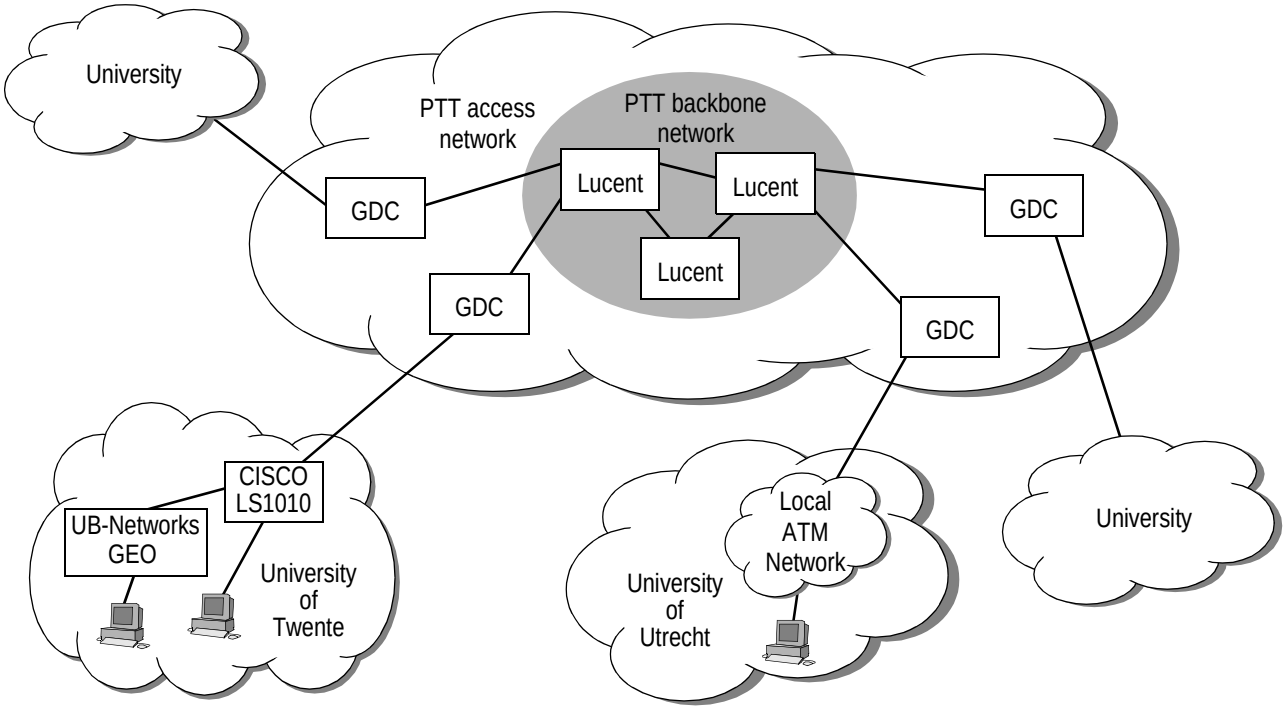

Figure 1: The SURFnet ATM network

Section 2 discusses the design of our Customer Network Management (CNM) tool. The tool is based on a client-server approach and uses web technology. Section 3 describes the architecture of the server and section 4 the architecture of the client. Section 5 provides the conclusions.

2 Design

The design of the management tool will be explained in four sections. Section 2.1 discusses the overall design approach and motivates the choice for a client server approach. Section 2.2. considers the consequences of the multi vendor aspects of SURFnet. Section 2.3 discusses the fact that the tool should be useable by managers who are not experts in ATM. Finally section 2.4 discusses the security related issues with respect to the deployment of the tool.

2.1 Client Server Approach

The tool is designed for network managers within the various universities that are connected to SURFnet's ATM service. To make it easy for managers to use the tool, the designers decided to avoid the need to install special software within the manager's system. Since the World Wide Web is nowadays in widespread use, it is safe to assume that most network managers already have a web browser installed. To exploit the availability of these browsers, a client server approach has been selected in which the client part will be implemented as HTML pages and / or java applets.

One of the requirements of the tool is to offer an advanced user interface with graphical features. Such interface can be realised as a set of HTML pages with embedded GIF or JPEG images that are generated by the server. Generating such images for multiple clients can result in a heavy server load and in large amounts of network traffic. In addition, every interaction between the user and the tool results in an interaction between the client and the server, and increases server and network load even further. As a result, response times will be bad.

To overcome these problems, the decision was made to implement the client as a java applet with local intelligence and the capability to generate graphs.

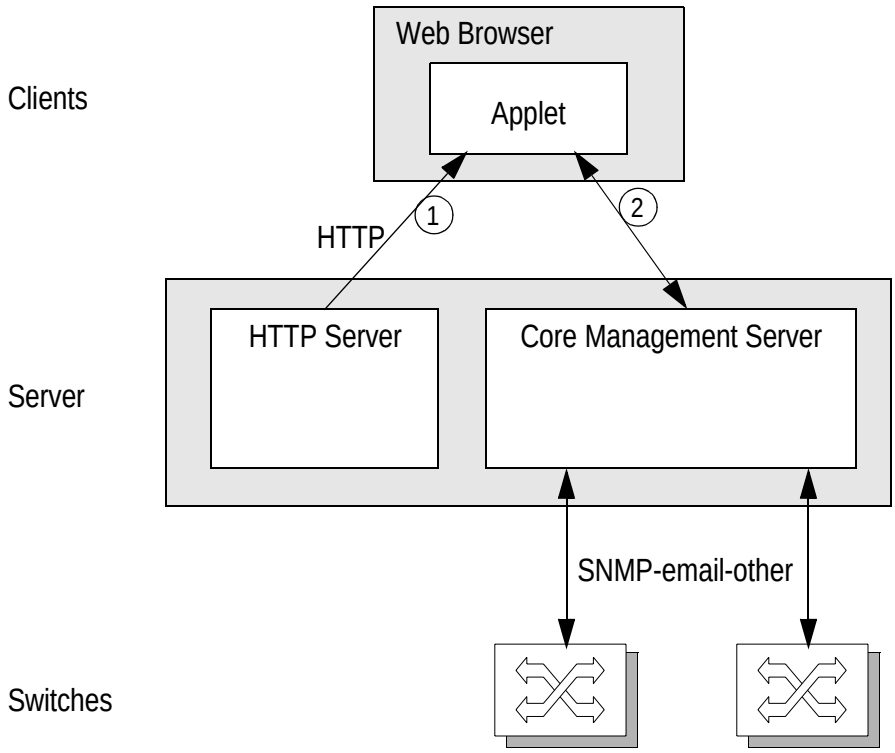

Figure 2: Client Server Approach

The server system consists of two separate processes; a standard HTTP server process which can be accessed by a web browser and the Core Management Server process which provides the actual ATM PVC management functionality. The client connects to the server in two steps:

1) The client uses a web browser to download a web page containing the applet from the HTTP server running on the server system.
2) The applet creates a separate connection to the Core Management Server process running on the server system. This connection is implemented using sockets and a proprietary protocol, since other communication mechanisms like Remote Method Invocation or CORBA were not available in JDK 1.0.2 [7], which is the java version used in this project.

2.2 Multi Vendor Capability

The Dutch research ATM network consist of switches from multiple vendors. Most of these switches can be managed via SNMP, which implies that the tool must support SNMP. Furthermore, switches from different vendors use proprietary MIBs, which implies that the tool must also support different MIBs.

To hide these differences from the tool user, it was decided to define a common management view for all ATM switches (see section 3).

2.3 High Level of Control

The user interface of the management tool must be designed in a way that both experienced users and users without in-depth knowledge of ATM will be able to work with the tool. As a consequence, the traffic parameters that must be specified by the manager to create new cross-connects will be presented in an intuitive way, with a set of default values that may satisfy most cases.

In the ideal case end-to-end PVCs should be configurable with a single command. The current version of the tool does not (yet) provide this capability, but requires the configuration of every switch along the path of the PVC. To delete a PVC, again each switch on the past must be configured separately.

2.4 Security

Security issues apply for all tasks that are to be performed with the management tool:

- *Authentication* is required to verify the identity of the users of the tool. Without authentication it would be possible for users to impersonate other users and in that way get unauthorized management access to ATM switches.
- *Encryption* is required to prevent information that is transported over the network to be exposed to others, for example the passwords that users must supply to gain access.
- *Access control* is required to allow only selected users to perform certain management functions. As an example, any user can get an overview of existing cross-connects while only specific network managers are allowed to create new cross-connects.

Another security issue to be addressed relates to the fact that the Dutch PTT does not allow anyone outside the PTT to directly access the switches in their backbone and access networks. A solution to this problem is to model the entire PTT network as a black box, and consider this black box to be a *distributed virtual switch.* The cross-connects within this virtual switch directly connect the 'outside ports' to the various universities. From the user's point of view, setting up the PTT portion of a PVC is thus reduced to creating a single cross-connect in the virtual switch.

To add virtual switch support to the tool, a special driver has been developed. This driver does not access any switches in the PTT network directly, but sends an email messages to the PTT network manager instead. This email contains all information necessary to create or remove a cross-connect in the distributed virtual switch.

3 Core Management Server

The Core Management Server consists of two layers: the client registration layer and the driver layer (Figure 3).

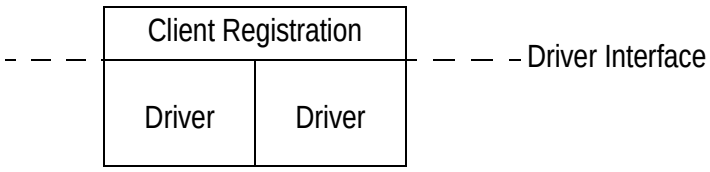

Figure 3: Structure of Core Management Server

The Client Registration part of the server implements the actual communication with the client applets. The drivers communicate with the various ATM switches.

3.1 Client Registration

The client registration layer keeps state information on all clients that are currently connected to the server. The communication between client and server is connection oriented; the server permanently keeps a separate socket connection open for each connected client. The server and client use a proprietary protocol to communicate over this socket: java objects send their information via the socket connection and at the receiving end the object is reconstructed. Transport over this socket can be encrypted using the *Secure Socket Layer* [9].

3.2 Driver Interface Specification

The driver interface (Figure 3) specifies the set of management functions that any ATM switch must support. This specification is defined at an abstraction level where the differences in protocols and MIBs supported by the various switches are no longer visible. Clients send requests for invocations

of these general management functions to the server together with a parameter specifying for which of the available switches the function invocation is intended.
The Driver Interface in the Core Management Server includes the following functions:

- *Get System Information* This function gives general information about the selected switch, including the elapsed time since the switch was last rebooted, the physical location of the switch and who to contact in case of problems with the switch.
- *Port Statistics.* This function presents an overview of the port statistics of the selected switch. This overview includes the number of transmitted and received octets, and the number of received errored cells.
- *cross-connects*. The cross-connect functions have parameters to specify endpoints as well as parameters to specify traffic descriptors. An endpoint is specified by the combination of its port number, VPI value and VCI value. A traffic descriptor has the following parameters:
  - The traffic class; this can either be Unspecified Bit Rate (UBR), Constant Bit Rate (CBR) or Variable Bit Rate (VBR).
  - The Peak Cell Rate (PCR) in cells per second.
  - The Sustainable Cell Rate (SCR) in cells per second
  - The maximum burst size in cells.
  - Whether or not tagging is used. Tagging is the act of modifying the Cell Loss Priority (CLP) bit of ATM cells in case of traffic descriptor violations.

  The peak cell rate, sustainable cell rate and burst size parameters can be specified twice; once for the guaranteed cell stream (cells with CLP=0) and once for the total cell stream (cells with CLP=0+1).
  The three functions for cross-connects are:
  - *Overview*. This function provides an overview of all the currently configured cross-connects. It supplies for each cross-connect the VPI, VCI and port numbers of both endpoints, as well as the traffic descriptors for each direction of the connection.
  - *Create*. This functions creates a new cross-connect. Parameters to this function include the VPI, VCI and port numbers of both endpoints, as well as the traffic descriptors for each direction of the connection.
  - *Destroy.* This function destroys a specific cross-connect. Parameters to this function include the VPI, VCI and port numbers of both endpoints.

3.3 Drivers

The drivers of the Core Management Server convert the vendor independent functions of the driver interface into vendor specific commands for the various kinds of switches. In the opposite way the drivers translate the information that is extracted from the switches into the vendor independent driver interface format.
For the current version of the management tool two drivers have been implemented:

1) A driver that generates email messages. This driver is used to manage the distributed virtual switch of the PTT. The driver converts requests to establish and release cross-connects into email messages, but does not support the flow of information from the PTT back to the management tool.
2) A SNMP AToM MIB driver which uses the SNMPv1 [2][8] protocol to access switches that have implemented the AToM MIB [1]. This driver translates the functions at the driver interface into (a set of) SNMP *SET* and *GET* messages for the AToM MIB. Figure 4 shows the mapping from driver interface functions onto the AToM MIB objects.

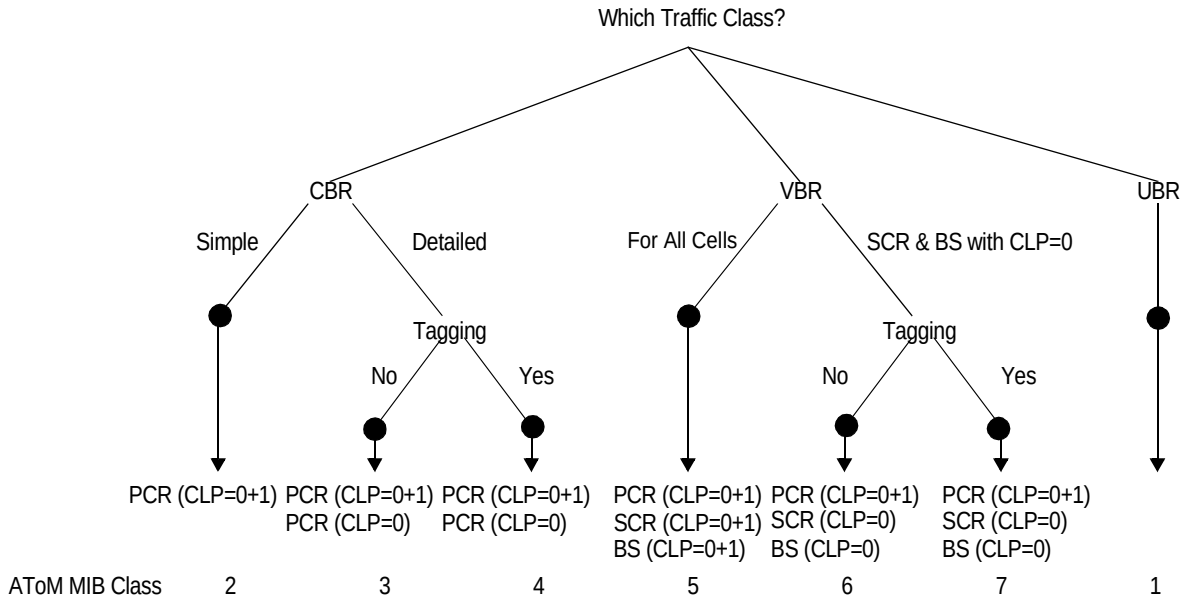

Figure 4: Mapping from driver interface functions onto AToM MIB objects

4 Client

The client side of the management tool is implemented as a java applet which runs on the java virtual machine of a web browser. The main task of the applet is to provide an easy to use graphical user interface. Section 4.1 discusses this user interface; Section 4.2 discusses the access control mechanism implemented by the client.

4.1 The user interface

The user interface of the management tool provides similar functions as the ones defined at the driver interface (Section 3.2). The three information presenting functions, *'get system information'*, *'cross-connect overview'* and *'port statistics'*, generate tables. The two other functions have a more complex interface. The *'destroy cross-connect'* function presents a list of all existing cross-connects, and allows the user to choose from this list the cross-connect to be destroyed. The *'create cross-connect'* function is the most complicated function, and will be discussed in the remainder of this section.
The process of creating a cross-connect has been divided into two steps:

1) The first step is to ask the manager to specify both endpoints of the cross-connect. Each endpoint is defined by a port number and VPI/VCI combination. To be able to specify for both directions of the connection different traffic descriptors, the endpoints will be named A and B. Using these names, both directions can be uniquely identified: from A to B or from B to A. Figure 5 shows the user interface that belongs to this step.
2) The second step is to specify for each direction a traffic descriptor. In most cases the manager will select one of the default values: *'audio'*, *'low quality video'*, *'medium quality video'*, *'high quality video'* or *'unspecified'* (for UBR traffic). However, the manager may also provide custom values. The parameters that can be specified for this purpose resemble the traffic descriptor parameters that have been defined by the AToM MIB [1]. Since the semantics of these parameters are difficult to understand, the tool provides the manager with an easy to use graphical interface. First of all the manager selects one of the traffic classes defined by the ATM Forum [6]. Currently the tool supports CBR and VBR, UBR has no traffic descriptors. Subsequently the tool provides the user with a picture that explains in an intuitive way the semantics of the parameters that need to be specified (Figure 6). Depending on the selected parameter values, other parameters may show up.

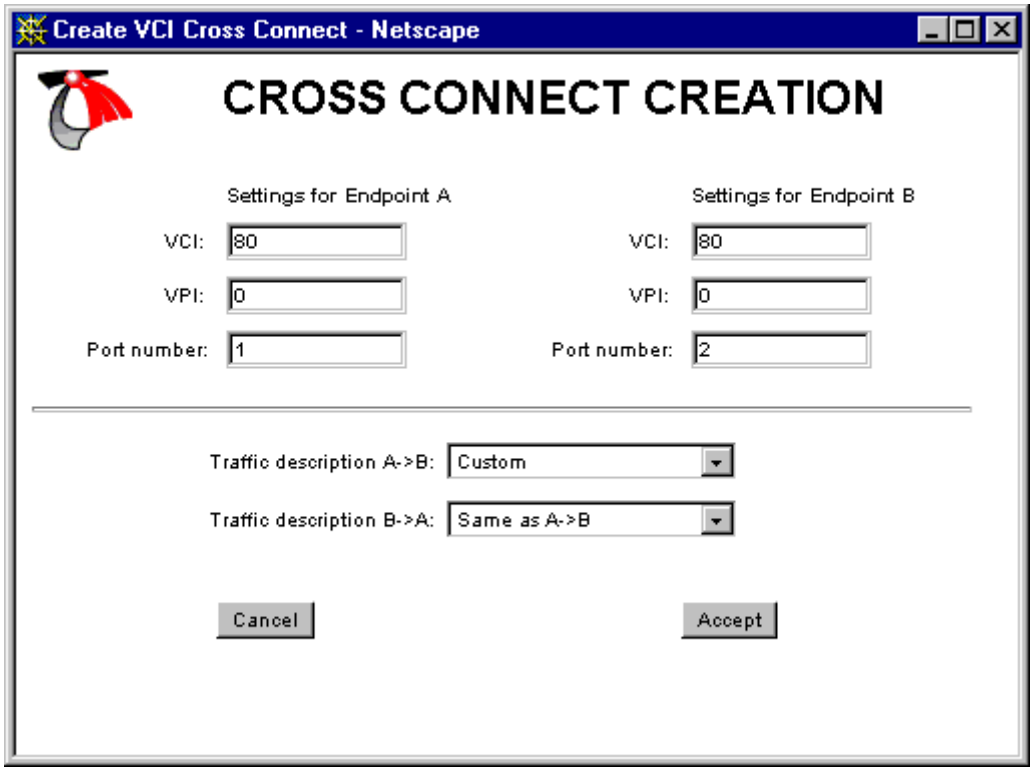

Figure 5: Create a cross-connect user interface

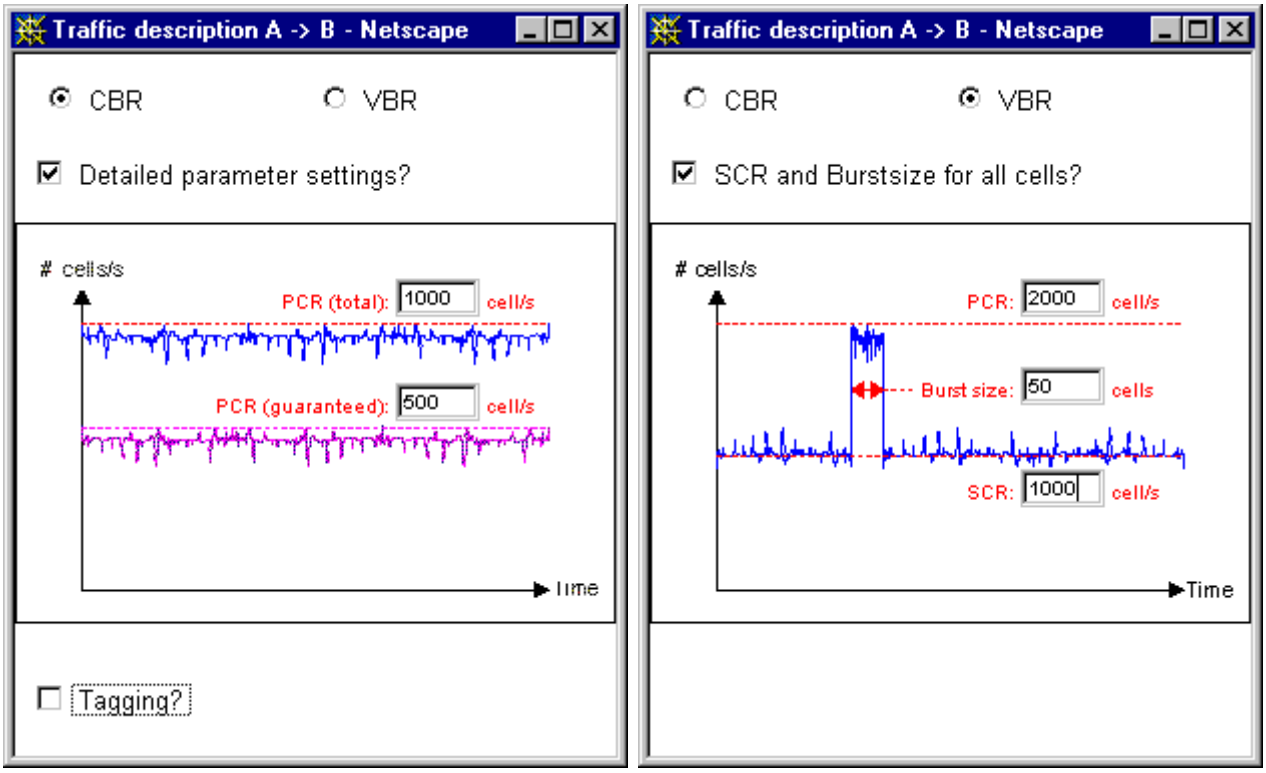

Figure 6: The user interface for the CBR and VBR traffic descriptors

The CBR screen has a check-box called *detailed parameter settings*. If this check-box is off (the normal mode) only the PCR parameter appears. If the check-box is on (the detailed mode) separate PCR parameters can be specified for the guaranteed cell stream (cells with CLP=1) and the total cell stream (CLP=0+1). In this mode the user can also specify to tag guaranteed cells that violate the traffic contract. If tagging is not selected, violating cells will be dropped.

The VBR screen contains parameters for the PCR, the burst size and SCR. The burst size and the SCR can be specified for the total cell stream (CLP=0+1), or only for the guaranteed cell stream (CLP=0). In this last case, the manager can also specify whether tagging has to be performed.

4.2 Access control

In the current version of the tool access control is implemented via a password based *log-on* mechanism. Access control is particularly needed for the *create* and *destroy* functions, since these functions make actual changes to the operation of the switch. The other functions (*system information*, *cross-connect overview* and *port statistics*) provide only information and do not make any changes to the switch. Since the current version of the tool is often used for demonstration purposes, usage of these functions is not (yet) protected. If this protection is needed, the basic authentication mechanism of HTTP [10] can be used to protect the home page of the management tool with a password.

5 Conclusions

In this paper the design of a web based ATM switch management tool has been discussed. The tool is intended for the network managers at the universities that are connected to the ATM research net in the Netherlands. The tool enables managers to set-up PVCs between their own ATM domain and the ATM domains of other universities. The following requirements played a central role in the tool design:

- *Assist in the establishment of PVCs*. The tool enables managers to create and delete PVCs on a per switch basis. For each switch on the PVC's path the manager can get an overview of the currently configured cross-connects (the part of the PVCs that goes through the switch), set up new cross-connects or destroy existing cross-connects. The manager can also obtain a statistical overview.
- *Support switches from multiple vendors*. The tool is designed in a modular fashion with clearly defined interfaces. To manage switches from multiple vendors, different drivers should be loaded. Drivers map the vendor independent driver commands of the driver interface onto vendor specific management commands for a particular kind of switch. For example a SNMP driver has been developed for ATM switches with SNMPv1 access that implement the AToM MIB.
- *Provide a high level user interface*. The tool offers a high level graphical user interface and enables managers with little knowledge of ATM to set up and destroy PVCs. A lot of attention was paid to the user interface that allow the provision of traffic descriptor parameters in an intuitive and user friendly way.
- *Enable secure operation*. The tool prevents unauthorized users from having access to the management capabilities of the ATM network. In the ideal case the server side of the tool is installed within each network domain on a separate server system. Through this approach the server system acts as a service hatch between the web-browser of the manager and the switches within that domain. Since the client side of the tool is implemented as a java applet, the manager does not have to install special software. Communication between client and server can be based on the Secure Socket Layer.

The idea behind the current tool was to manage a single ATM switch at a time. To set up a complete end-to-end PVC, all intermediate switches on the PVC's path have to be configured. This is a time consuming and not very user friendly method. Future versions of the tool will therefore be enhanced to support the establishment of end-to-end PVC. Two options are possible. In the first option the tool creates a map of the network topology and selects the switches that should be used for the PVC. In the second option end-to-end PVCs will be created via a technique called *soft PVCs* [5].

The tool allows managers with little background in ATM to create and delete PVCs. The tool is available in the public domain, and can be downloaded from *http://wwwsnmp.cs.utwente.nl/nm/research/projects/utopia/*

6 Acknowledgements

The research described in this paper has been performed for and funded by SURFnet bv. In particular we would like to thank Victor Reijs, Niels den Otter and Ronald van der Pol for their contributions to this research project and for their comments and suggestions on this report.

7 References

[1] M. Ahmed, K. Tesink, *Definitions of Managed Objects for ATM Management Version 8.0 using SMIv2*, RFC1695

[2] J. Case, M. Fedor, M. Schoffstall, J. Davin, *A Simple Network Management Protocol (SNMPv1)*, RFC1157

[3] NM Group, *UTopia: The Web Based ATM Management tool*, http://wwwsnmp.cs.utwente.nl/nm/research/projects/utopia/

[4] W. Kasteleijn, *Thesis on Web based management*, ftp://ftp.cs.utwente.nl/pub/src/snmp/UT-THESIS/Kasteleijn.ps

[5] The ATM Forum, *Private Network-Network Interface Specification Version 1.0 Addendum (Soft PVC MIB)*, af-pnni-0066.000 September 1996

[6] The ATM Forum, *The ATM Management Specification version 4*, af-pnni-0056.000, April 1996

[7] Sun Microsystems inc., *The JAVA Developers Kit 1.0.2*, http://java.sun.com/products/jdk/1.0.2/

[8] Advent, *Java SNMPv1 stack*, http://www.adventnet.com/

[9] Netscape, *The SSL Protocol*, http://home.netscape.com/newsref/std/SSL.html

[10] T. Berners-Lee, et al, *Hypertext Transfer Protocol -- HTTP/1.0*, RFC1945