Integration of Mobile Agents and Legacy System to Manage ATM Heterogeneous Networks

André Ribeiro Cardoso

Universidade Federal da Paraíba - UFPb
Rua Aprígio Veloso, 882
Campina Grande, Paraíba, Brazil.
andrec@uece.br

Joaquim Celestino Júnior, RicardoA. R. Celestino

Universidade Estadual do Ceará - UECE
Av. Paranjana 1700,
Fortaleza, Ceará, Brazil.
{celestino, ricardo}@larces.uece.br

ABSTRACT. *This paper presents a research about the mobile agents technology integrated with legacy systems of the "client/server" technology. Because of the device growth connected to the Internet we identify some problems related with the "client/server" technology behavior, the centralized network management and the PVC tasks in ATM networks. Based on the integration among mobile agents and legacy systems we proposed our solution to these problems. Thus mobile agents interaction with SNMP agents are employed to create a heterogeneous method to establish PVC in ATM networks. At LARCES/UECE we developed a PVC manager u sing this integration. As a first step we used Concordia system as mobile agent platform.*

KEYWORDS: Network Management, Mobile Agents, Legacy System, SNMP, MIB, Concordia.

1. Introduction

Nowadays a lot of applications are launched to run on the Internet . Therefore, it is substantial the number of servers being connected to the web that uses the “client/server” architecture, greatly employed in the network.

Technologies such as Asynchronous Transfer Mode (ATM) were presented with the goal to increase the information transfer speed. Widely employed in this technology, end-to-end connections, called Permanent Virtual Circuits (PVCs), are complex tasks due to the peculiarities of each vendor device.

Because of the device growth connected to the Internet, which is greatest motivation of this research, we identified some problems related to the “client/sever” paradigm behavior, the network management and the PVC tasks in ATM networks. Hence we propose our solution.

Related to the paradigm, as shown above, the “client/server” architecture has become a bottleneck due to the increasing number of traffic information caused by the “request/response” created in this model. It is evident that this architecture became inefficient in these aspects and needs some changes to improve its efficiency.

Nowadays, existing network management applications are based on one of the two protocols to address the problem of interoperability in heterogeneous environments: Simple Network Management Protocol (SNMP) (Stallings,1998) for data networks and Common Management Information Protocol (CMIP) (CCITT X.711,1991) for telecommunication networks. Unfortunately, these protocols are based on static and centralized “client/server” solutions, where every element of the network sends all the data to a central location that processes the whole data and provides the interface to the other operator. Because of that, management applications have scalability problems and produce too much traffic in the network.

Finally, related to the heterogeneous PVC process, it is well known that PVC is considered one of the most important tasks of ATM management network. There is no standard or uniform way for the network operator to set up PVC in a heterogeneous network. For example, setting up an end-to-end PVC among different places requires the configuration of all the intermediate switches on the path. Because the switches come from multiple vendors, each one with a proprietary management interface, it is inconvenient to set up such PVC. Thus it is clear that setting up PVC is a complex task that needs tools to make it easier.

This research was based on the project “A CORBA Distributed Platform with Intelligent Mobile Agents for Service Management (AMI)” of the program “High Speed Metropolitan Network of Fortaleza (REMAV-FOR”). At “*Laboratório de Redes de Computadores e Engenharia de Software da Universidade Estadual do Ceará* (LARCES/UECE)” we developed a research for PVC configuration exploiting the integration among mobile agents and legacy systems, presented in the final part of this paper.

2. Related Work

Several aproaches have been developed dealing with mobile agents, network management, and PVC configuration. We follow their main idea to develop our research. SNMP integrated with mobile agents are highly considered in our proposal to create a heterogeneous way to establish end-to-end connections.

Asynchronous Message Transfer Agent System (AMETAS) (Zapf *et al*,1999) shows the integration of two management networks paradigms: legacy management protocol such as SNMP with the decentralized management of complex network using mobile agents. The key idea behind this is the employment of SNMP mainly for collection data locally instead of remotely. The exchange of a large number of messages between the manager and the SNMP agent over the network is replaced by the migration of a SNMP-enabled AMETAS agent, which applies SNMP locally. Thus, the network load produced by management applications can be largely reduced. Based on AMETAS, there is a network management framework - NetDoctor that allows the monitoring of hosts in a heterogeneous network. This framework presents mobile agents that may query local or remote SNMP agents, process the status data, and perform corrective actions. Our research explores this idea of interaction with SNMP data locally by the mobile agents importing some Java classes for this aim.

Perpetuum Mobile Procura project (Bieszczad,1997), from Carleton University, developed an infrastructure that provides a framework for code mobility. Perpetuum Mobile Procura proposed an approach that uses SNMP-DPI protocol adapted and implemented in Java, in a symmetric way, for achieving full bi-directional SNMP type interaction between mobile agents and legacy DPI enabled SNMP agents (Pagurek e*t al.*,1998). This is also a project where SNMP is used both to access local management services and to interface with legacy management applications. According to this project, methodologies for PVC configuration in heterogeneous ATM environments were developed in (Boyer *et al.*, 1999). Based on the main idea of its methodologies we proposed a research at LARCES/UECE where Concordia mobile agents establish a PVC through REMAV devices.

A CORBA Distributed Platform with Intelligent Mobile Agents for Service Management (AMI) project main objective is to explore the technology of mobile agents and its applications connected to the management of network and services. As a first step, it proposes a framework based on the technology already in use, like the Concordia that is used in this paper, and afterwards the construction of our own platform to execute mobile agents developed at LARCES/UECE.

AMI project suggests the following framework divided in five levels:

- Application of Service Management Based on Mobile Agents
- Service of Mobile Agent Support: Agency
- Distributed Support: CORBA and RMI API of Java
- Proxy
- Network Element

From AMI project, initially we are creating Concordia mobile agents to configure PVCs (Cardoso *et al.*, 2001) through the devices of REMAV in Fortaleza and afterwards the devices of other REMAVs in Brazil.

3. ATM Background

Few technologies have been adopted with such enthusiasm as ATM. In fact, ATM is emerging as a great and promising network technology due to its velocity, scalability, flexibility and the guarantee of quality of service (QoS). ATM offers a good combination of switching packed circuit technique.

The technology ATM uses cells of fixed sizes of 53 bytes. There are the Virtual Path Identifier (VPI) of 8 bytes and the Virtual Channel Identifier (VCI) of 16 bytes. VPI and VCI are the only cells that belong to the same Virtual Connection on a shared transmission medium. ATM operates in a connection-oriented model. Before the cells are transmitted from one user to the other, a phase to establish a logical/virtual connection allows the network to reserve the necessary resources, such as bandwidth. There are two kinds of mechanisms to establish a connection: Permanent Virtual Circuit (PVC) and Switch Virtual Circuit (SVC). The first is pre-established at each device along the network and the second one is established under demand, based on procedures of signaling.

3.1. *Using SNMP to Manage ATM Networks*

Two important standardization organizations are involved in standardizing management of ATM network using SNMP protocol for the transport of management information. They are Internet Engineering Task Force (IETF) and the ATM Forum. This paper deals with the first one, ATM management standards of IETF.

The Internet-standard network management framework, known as SNMP has reached good results in providing interoperable solutions to the problem of network management by enabling effective monitoring and control of heterogeneous devices. Today, SNMP is widely used in network management. Nowadays, there are three versions of SNMP management systems: SNMPv1, SNMPv2 and SNMPv3.

Three requirements have to be fulfilled to make an ATM network manageable through SNMP (Sprenkels, 1996):

- The devices must contain SNMP agents and a collection of management information, named MIB.

- Each device is responsible for the changes in the system behavior, registered in its MIB. For example, consider a table of ATM connections in the MIB of an ATM switch. After a manager has created an entry in a table for a new connection, the switch software must create that connection. Registers are then modified in different parts of the hardware of the switch and the switch framework.

- A manager should be able to exchange SNMP Protocol Data Units (PDUs). This possibility to exchange SNMP PDUs between the manager and the agent entities is reached by the use of a management information transportation service.

4. Mobile Agents

Within the computing and information systems, the notion of mobile agents implies a program remotely running with a certain degree of autonomy that usually assists the tasks of processing and restoring information.

Mobile Agents introduced a new software and communication architecture by allowing a program to travel among different computers to run remotely, even among heterogeneous networks. The idea of remote performance through the transmission of executable codes among clients and servers has become more and more popular in recent years in the area of intelligent networks. In the transport of the agent code to other computers in distributed network, it is not necessary to carry intermediate data through the network which significantly widens the bandwidth and it can also avoids the delays of communication.

The concept of mobile agent was originated from three technologies: migration of processes, remote evaluation and mobile objects, all three developed to improve the Remote Procedure Call (RPC) for the distributed programming.

The mobile agent is a software entity in which there is a software environment. It inherited some characteristics of an agent.

5. Concordia

Mitsubishi Electric Information Technology Center America created the Concordia System (Concordia,1997), with the objective to develop, implement and manage mobile agents applications in order to access information, at any time, place and/or any device supporting Java.

5.1. *Concordia Components*

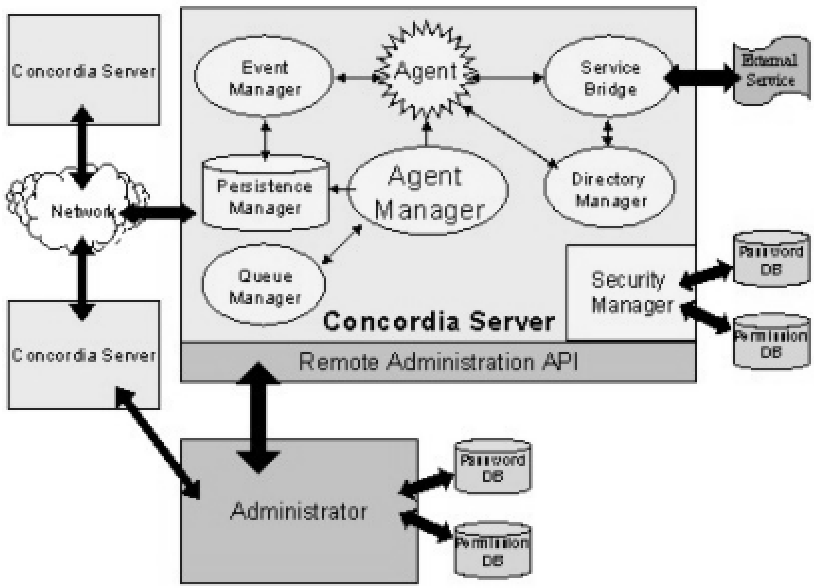


Figure 1. *Concordia Components*

Concordia contains multiple components written in Java that together provide a complete framework for mobile agents. Concordia Server is the biggest block in which reside various Concordia managers. The components of Concordia are the following (Figure 1)(MAC-WhitePaper, 1998):

Concordia Server is a framework that offers an infrastructure for the development and management of network mobile agent applications. Its design goals focus on providing coverage of flexible agent mobility, support for agent collaboration, persistence of agent state, reliable agent transmission, and agent security.

The Agent Manager component serves as the communication server for agent transfer. An agent program initiates its transfer by invoking the Agent Manager methods. The Agent Manager sends the agent to another Agent Manager of another Concordia Server. Concordia extends the agent interaction to support two forms of inter-agent communication: asynchronous distributed events and agent collaboration. Asynchronous distributed events are scheduled and managed by the **Event Manager** or **Inter-Agent Communication Manager** component, while agent collaboration requires the agent application programmer to specify a collaborating object through the utilization of the Concordia class libraries. Agent collaboration allows agents to interact, modify external states, as well as internal agent states.

Persistent Storage Manager was designed and developed to support persistence and recovery of agents from system crashes. It may also be used to store internal state to facilitate recovery after server failure. Its implementation is based on the Java object serialization facilities.

The Queue Manager component was developed to relieve potential performance and reliability problems associated with the transmission of agents across networks with different characteristics in the communication medium. The Queue Manager component manages inbound and outbound queues for the reliable transport of agents across the network.

Concordia security model provides support for two types of protection: protection of agents from being tampered with (called agent protection), and protection of server resources from unauthorized access (called resource protection) that addresses the problem of a host being attacked or misused by agents. Agents are protected from tampering while stored on client systems during transmission and while stored on the persistence store. **The Security Manager** (Walsh *et al.,* 1998) component is a Java object that manages resource protection.

The Administration Manager component handles the Concordia system administration. It starts up and shuts down other servers in the Concordia agent system. The Administration Manager also monitors the progress of agents throughout the network and maintains agent and systems statistics.

Name service is done by the **Directory Manager** component. Therefore, mobile agents are enabled to locate the application servers that hope to interact with one host.

Agents Tool Library is the group of tools and necessary classes that allows the development of Concordia mobile agents.

6. Our Proposal

The main goal of this work is to integrate mobile agents with legacy systems. This way, we can solve the problematic related in the beginning of this paper.

6.1. *For the Paradigm*

Mobile agents overcame some limitations of the “client/server” paradigm. With mobile agents, the problem of robust networks is greatly decreased because the hold time for connections is reduced to only the time required to move the agent in or out of the device. The classic existing agents in network systems tend to be monolithic.

Mobile agents do not statically reside on network devices, therefore can be created on demand and destroyed when no longer required. They tend to be smaller then the agents in classic network management systems because they normally perform a single task. In general, mobile agents can reduce the load on the manager side because a large management task can often be divided into smaller tasks delegated to such mobile agents.

6.2. *For the Network Manager*

Network management based on the SNMP and CMIP leads to a centralized management architecture. The use of polling-oriented mode is not advised for increasing networks because of its drawback in respect to performance, scalability and inefficiency. Management by delegation and mobile management agents are attempts to introduce decentralized management architecture. Although these new management approaches seem to solve the problems of centralized management architectures, we cannot underestimate the power of existing management solutions based on legacy protocols such as SNMP. SNMP is an accepted standard, widely supported by commercial products. Therefore it is very interesting to try to integrate legacy systems management with the new management proposals.

The issue of interoperability with existing management technologies, like SNMP and CMIP is very important for the success of mobile agents in the network management area. In order to ensure the development of applications for the management of large and heterogeneous systems, mobile agents systems need to provide interoperability.

First, because the legacy technology provides access to management information and services. Thus it is very interesting that mobile agents incorporate mobile code into the existing local management services, in order to perform intelligent tasks closer to management data. The employment of SNMP is mainly to collect data locally instead of remotely. Therefore, the mobile agents will not replace the classic protocols used to manage applications in heterogeneous network. Instead, they will complement them with powerful programming constructions allowing more efficient solutions for network management.

Second, mobile agents management applications need to coexist with legacy management systems. They are well suited to develop new network management services, although it seems much more attractive to use these services from installed management applications than to develop separate specialized applications. This way, mobile agents can be introduced to solve specific problems for larger management frameworks still based on classic paradigms.

We can draw conclusions that mobile agent application solutions present profits in situations where it is appropriate to migrate code to the data instead of sending the data to application. To overcome problems such as scalability, flexibility and performance, by dividing management functions into mobile, autonomous and intelligent computing entities, many problems with the network and service management can be solved:

- Scalability is increased, when management is not performed solely by the management station but delegated to distributed management agents;
- Better performance is achieved by moving the management functionality closer to the actual network element, thus reducing network traffic;
- The low -level details of different devices can be hidden behind the agent interface;
- Legacy systems can be integrated by using an agent for inter-operation.

6.3. *For the PVC Heterogeneous PVC Process*

PVC operations and configuration are the principal PVC management tasks in ATM networks. The heterogeneous and proprietary management provided by ATM switch vendors has created an environment where it is very difficult to automate the process. The management situation will become more and more unpredictable as networks continue to grow in size and complexity.

A network management system consists of four parts: a management station, an agent, a management information base (MIB), and network protocols. Due to the nature of design differences, each vendor currently develops its device MIB. ATM Forum and IETF tried to define standard MIBs such as ILMI (ATM ilmi - 0065.000,1996) and AToM MIBs (Ahmed *et al.*,1992) respectively.

They covered public and private User to Network Interfaces. But AToM MIB is more oriented towards PVC configuration and so we selected it in this work. RFC 1695 defines a MIB used to manage ATM based interfaces, devices, networks and services. Its primary goal is to manage ATM PVCs. The AToM MIB basic structure is: Interface Information, Virtual Link Information and Cross -Connect Information (Figure 2).

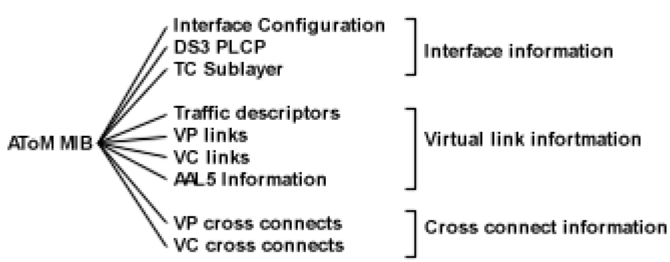


Figure 2. *AtoM MIB*

In order to integrate the facilities of SNMP legacy management with mobile agents technology to complement the “client/server” architecture we created a heterogeneous methodology to establish PVCs in ATM networks, overcoming the constraints of the existing differences among ATM device vendors. Therefore, a simple and generic way to perform this management task is proposed by hiding the peculiarities of each ATM device system.

7. System Prototype

In order to analyze the solution of mobile agents providing the functionality of PVC configuration, a prototype of Concordia was developed, offering a general view of PVC tasks in ATM devices (hosts and switches).

We needed to make some assumptions and requirements in order to simplify the implementation without damaging the main idea of the research.

The necessary assumptions are:

1- the functionality of the process is only related to the configuration of point-to-point PVCs. Therefore we do not consider SVCs and multipoint PVCs;

2- the QoS parameter evaluated is Peak Cell Rate (PCR), which is considered as the maximum bandwidth allocated to a connection;

3- the ATM service class considered is ‘best effort’ service;

4- the user has the knowledge of the whole environment (hosts and switches) along the connection path, meaning then that the route is predefined and no decision about the routing should be made.

5- the Concordia System evaluation version 1.1.7 of 2001 is employed and because of that some mobile agent security issues are not considered.

The requirements are:

1- all devices (hosts and switches) must be manageable with SNMP through AtoM MIB;

2- the user has an interface applet to launch the PVC configuration mobile agents;

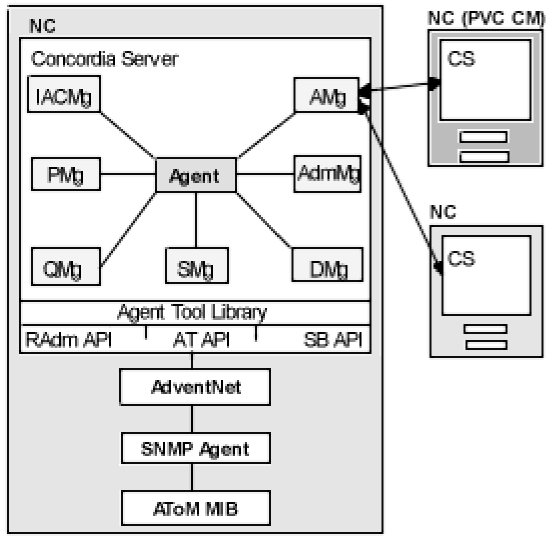
3- a set of mobile agents to perform PVC set up;

4- a set of mobile agents to perform PVC trace;

5- a set of mobile agents to perform PVC release.

The architecture of the system is composed by the components defined below. All mobile agents in the prototype are implemented by using Concordia (Figure 3). In the Concordia Infrastructure, agent application programs are implemented as Java objects. Therefore users would first need to write a Java class that specifies the action desired. Once this Java class is written and compiled, the user can launch the agent program in three ways: via GUI Agent Launch Wizard, via command line tool, or using the external API.

7.1. *Implementation Architecture*



NC: ***Network Component***
IACMg: ***Inter-Agent Communication Manager***
AMg: ***Agent Manager***
AdmMg: ***Administration Manager***
RAdm API: ***Remote Admin API***
SB API: ***Service Bridge API***
PMg: ***Persistence Manager***
SMg: ***Security Manager***
QMg: ***Queue Manager***
DMg: ***Directory Manager***
ATAPI: ***Agent Transport API***
PVC.CM:***PVC Config.Manager***

Figure 3. *Implementation Architecture*

The Concordia Server must be present in each device since it is a mobile agent framework on various platforms. In case the switch does not execute the JVM, the components of the system must reside in another NC that is executed in a separate host responsible for the management of its resources. The Concordia Server provides the necessary intelligence to configure an ATM network. The mobile agents are implemented to execute the different PVC configuration tasks by using the functionality of ATM devices.

The PVC Configuration Manager component, re sponsible for the PVC management configuration tasks of the device is a Concordia mobile agent launcher developed using the last type – the external API – written to customize launch class which uses the Concordia class libraries. It injects mobile agents into the ATM network. It specifies the group of switches along the PVC path, besides initializing the VPI, VCI, bandwidth etc.

To interact with SNMP agent to manage low-level resources, it is possible to include full SNMP type capabilities in mobile agent that provides PVC. Hence, it can issue GETs and SETs to provide PVC by using some SNMP classes available in Java for this purpose. As a first step, we employ a group of classes from AdventNet (AdventNet, 2001), since it supports the JDK 1.1 and more recent JVM. This package enables us to access management data from every SNMP-capable device. Therefore, importing these classes does all Concordia mobile agents interaction with SNMP agents to access AtoM MIB.

AToM MIB contains objects with attributes and values associated to ATM (host and switch), defined according to SMI format. For the prototype used in this paper, the handled objects are the necessary ones for PVCs configuration.

7.2. *PVCs Configuration Tasks*

Our research may include switches from multiple vendors. These switches must be managed via SNMP, which implies that the tool deals with SNMP support. Furthermore, switches from different vendors use proprietary MIBs, which implies that our tool has to use a common view of a MIB. We adopted the AToM MIB from IETF. This way we intend to hide the differences from MIB switch vendors. The user interface applet of the PVC CM component must be designed in a way that both experienced users and users without knowledge of ATM will be able to work with the tool. As a consequence, the traffic parameters that must be specified by the user to create new cross-connects will be presented in an intuitive way. Mobile agents, launched by PVC CM component, accomplish the PVC establishment sequentially, in other words, the mobile agent executes the configuration task in the first host, and then it migrates to the switches (s1 through sn). After it configures the switches, it travels to the final host and completes the configuration.

In case the switch does not execute JVM, the system components must reside in another NC that is executed in another host responsible for the management of its resources. When conditions of recoverable errors occur, the reconfiguration is done through a sequence of negotiations among mobile agents and devices. The steps of a PVC configuration mentioned above can be find in (Cardoso *et al*., 2001).

8. Conclusion

For the time being, the tests included the LARCES/UECE devices (PCs with ATM 25Mb and switches 8285 and 8265 IBM). Afterwards we intend to spread out the PVC operations to the remained devices of the REMAV from our city Fortaleza and others REMAVs throughout Brazil.

We hide the differences from switches vendors employing the integration of mobile agents with SNMP common view of MIBs. Therefore the user does not have to worry about the system of each switch and it is able to delegate the responsibility of configuration to mobile agents. The IETF AToM MIB information has its access possible because of the AdventNet classes performing the part of proxy agents between mobile agents and SNMP agents. The mobile agents automated the PVC configuration tasks without the need of interventions by users in the decisions.

Although the security mobile agents are not considered in our implementation, using the full Concordia System version can solve these issues. As it was demonstrated above, some assumptions were made but this work focuses on a great number of relevant aspects to the project and implementation of real architectures of mobile agent systems.

9. Bibliography

AdventNet API v2c Release 3.3 URL= http://www.adventnet.com, 2001.

AHMED, M., TESINK, T., IETF, RFC 1695, "Definitions of Managed Objects for ATM Management, Version 8.0 using SMIv2", 1992.

ATM Forum, "Integrated Local Management Interface (ILMI) Specification Version 4.0" af-ilmi-0065.000, 1996.

BIESZCZAD, A., "Advanced Network Management in the Network Management Perpetuum Mobile Procura Project", Technical Report SCE-97-07, Canada, 1997

BOYER, J., *et al*., "Methodologies for PVC Configuration in ATM Heterogeneous Environment using Intelligent Mobile Agents", MATA'99, Ottawa, Canada, 1999.

CARDOSO, A., CELESTINO JR, J., "A Proposal of PVCs Configuration Methodology Employing Mobile Agents", MATA'01, Montreal, Canada, Springer Verlag Pub., 2001.

CCITT Recommendation X.711, "Common Management Information Protocol (CMIP) Specification", 1991.

Concordia System Page URL=http://www.concordiaagents.com, 1997.

Mobile Agent Computing – A White Paper URL=http://www.concordiaagents.com/MobileAgentsWhitePaper.html, 1998.

PAGUREK B., *et.al*. "Configuration Management in Heterogeneous ATM Environments Using Mobile Agents", IATA'98, Paris, France, 1998.

SPRENKELS, R., "Management of ATM Networks", M.Sc-thesis, 1996.

STALLINGS, W. "SNMP, SNMPv2, SNMPv3 and RMON1 and 2", Addison Wesley, 1998.

WALSH, T., *et al*., "Security and Reliability in Concordia", HICSS31, USA, 1998.

ZAPF M., *et al.*, "Decentralized SNMP Management with Mobile Agents" IM '99, Boston, 1999.