A Proposal of PVCs Configuration Methodology Employing Mobile Agents

André Ribeiro CARDOSO

Coordenação de Pós-Graduação em Informática
Universidade Federal da Paraíba, Av. Aprígio Veloso, 882
Campina Grande, Paraíba, Brasil
E-mail: andrec@uece.br

Joaquim CELESTINO Jr

Departamento de Estatística e Computação
Universidade Estadual do Ceará, Av. Paranjana, 1700
Fortaleza, Ceará, Brasil
E-mail: celestino@uece.br

Abstract. This paper presents a research about the technology of mobile agents which has Concordia system as its platform. Concordia was developed by Mitsubishi Eletric Information Technology Center America. In the latest years, the Internet became the main media to access information, data and personal communication. As a result, the overload of information in the bandwidth has become inevitable. Because of the continuous increase of the server connected to the network, the present architecture "client/server" used to connect the computers has become inefficient, therefore relevant changes have been necessary. The development of the technology of mobile agents is considered an alternative solution, in which programs (agents) can move throughout the network to run in different computers. The Concordia's components were written completely in Java, and because of that, Concordia offers the portability and execution of its agents anywhere and at anytime with security, mobility, management and monitoring - all of the necessary characteristics to achieve the perfect environment for the mobile agents. 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"). In LARCES/UECE we developed a methodology for PVC configuration using mobile agent, presented in the final part of this work.

1 Introduction

With the growing number of servers connected to the web, the architecture "client/server", used to connect the computers, has become inefficient and because of that it needs great changes. The use of temporary solutions only transfers the information congestion problem to the Web.

Therefore, it is essential and urgent the development of new technologies to operate and manage the connection among various nodes on the Web. A possible solution for these problems consists in utilizing mobile agents that assist the user to perform his tasks. These agents can move to the location where data is stored and, with intelligence, select the information that the user needs, saving time, money and bandwidth.

Many programming languages are being developed and implemented for mobile agents framework. Nevertheless there are many problems related to its security that deserve special attention. This new framework needs, consequently, special attention to the mobile agent visits to agents systems, originated from other locations. Before allowing this type of access, for instance, we must guarantee the inaccessibility of certain services of the machine by the visitor agent, avoiding this way the excessive consume of its resources.

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"). In LARCES/UECE we developed a metodology for PVC configuration using mobile agent, presented in the final part of this work.

2 Framework of the AMI Management Service

AMI project main objective is to explore the technology of Mobile Agents and its applications connected to the management of network and services. At first, it proposes a framework based on the technology already in use, like the Concordia that is used here (in this paper), and afterwards the construction of a platform to execute mobile agents developed here at LARCES.

In our project we suggest the following framework divided in five levels:

1. <u>Application of Services Management Based on Mobile Agents</u> is composed by a group of intelligent mobile agents that supply the service management. It is made of a group of mobile agents that are dynamically developed and interact in a way to support the activities of service management. Different strategies of the mobile agent behavior and its cooperation can be introduced in this level.
2. <u>Service of Agent Support</u> contains a group of services that gives support to the mobile agent execution. Services of name, location and security should exist in this level. It should also contain some basic functions which allow the mobile agent to interact with local services. This layer integrates the specifications MASIF of OMG and adapts itself, when necessary, to give support to the network management and services.
3. <u>Distributed Support</u> will provide the functionality to support the interactions and mobility among mobile agents through the middleware CORBA - ORB and JAVA. This allows to isolate the upper layers of the subadjacents technology trough IDL interfaces. In this level , there is also a group of service support such as notification, persistence etc.
4. <u>Proxy Level</u> provides the necessary mechanisms to interact with the subadjacent agent . It provides the gateway mechanism to interact with SNMP/CMIP or legacy systems management.

5. Level of Network Management is made of a group of physical elements and software that are part of the service components and subjacent network. This level is based on the ATM infra-structure at LARCES.

3 Mobile Agents

Mobile Agents introduce 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 task of management delegation can be easily performed by mobile agents that can be reprogrammed. Mobile Agents can also access the remote resource of a device for specific management tasks.

The main difference between an intelligent agent and a traditional one is that the first one not only performs tasks pre-established by the user but also other tasks that can modify the environment. This characteristic is particularly useful when we are dealing with management when several times the following situation happens: the agents are a permanent part of the software that controls the managed entities. The policy of monitoring and controlling are left to the remote system of network management.

The concept of mobile agent was originated from three technologies: migration of processes [1], remote evaluation [2] and mobile objects [3] , all three developed to improve the Remote Procedure Call (RPC) for the distributed programming.

3.1 Mobile Agents and Management of Network Services

Services and network management is by its own nature a distributed activity that follows the model "client-server". In this model a central management entity controls all the network consisted of managed units. Many of the essential functions in the network management are performed in the model "client-server" while network entity with computing ability follows the philosophy proposed by Simple Network Management Protocol (SNMP) [4] of simple and passive agent structures. However, this approach has several technical limitations such as scalability, reliability, performance and difficulties in delegations through network that are becoming larger and badly distributed.

Network Management using delegations is an obvious alternative for centralized management. In a network management system there are delegated applications that work simultaneously as management units as well as managed agents. This delegation can be controlled and watched remotely. An efficient architecture of distributed management should address these important issues like reliability, flexibility, consistence and scalability.

4 Asynchronous Transfer Mode (ATM)

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 oriented connection 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.

A simple final system ATM or a switch does not support all the dimension end-to-end of a VC. Usually a VC is composed of multiple final and intermediate systems, each one supporting virtual links (VLs). Consequently, each final system supports a VC ending and the VLs in its external interfaces, whereas each intermediate system (switch ATM), through where a VC passes, supports multiples VLs in its external interfaces as well as the cross-connections of VLs belonged to the switch. The management end-to-end of a VC is reached through a combination of the management of its individual parts.

The VC is associated with a group of traffic descriptors specifying its characteristics, including the traffic parameters and the class of QoS. VLs inherent characteristics from the traffic of VC of which are part.

4.1 Network Management ATM

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) [5] and the ATM Forum [6]. This paper deals with the first one, ATM management standards of IETF where we will deal with the PVC parameters configuration established among final and intermediary systems of LARCES – UECE.

4.2 SNMP for ATM Management

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, SNPM 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[7]:

- 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.
- A manager should be able to exchange SNMP Protocol Data Units (PDUs).

AtoM MIB

The differences among the various versions of SNMP have a small effect in relation to the MIBs. The RFC 1695 [8] was developed to specify a MIB for the ATM network management. This MIB, also known as AtoM MIB, defines the object to manage ATM interfaces, virtual links, cross-connects, and AAL5 entities and connections supported by ATM hosts, ATM switches, and ATM networks. It complies with SNMPv2 SMI, and it is also semantically identical to the peer SNMPv1 definitions. Therefore it can be accessed by both the SNMPv1 and the SNMPv2 management applications.

The primarily purpose of the AtoM MIB is to manage ATM PVCs. Although ATM SVC information is also represented in the management information, full management of switched connections requires additional capabilities that are beyond the scope of the AtoM MIB. Each group of related objects is represented in this MIB as a conceptual table.

5 Concordia

Mitsubishi Electric Information Technology Center America created the Concordia System, 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

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. Some components have interface and, at any case, each one is responsible for a part of the project in a modular and extensible way. [9]

The components of Concordia are the following:

- *Concordia Server* is the name of the complete component installed and running on a machine in a Concordia Network;
- *The Agent Manger* provides the infrastructure of communication responsible for the transmission of agents;
- *Administrator Manager* provides the remote administration of Concordia;
- *Security Manager* protects the resources and guarantees the safety and integrity of mobile agents and their data;

- *Persistence Manager* maintains the state of mobile agents and objects in transit throughout the network;
- *Queue Manager* is responsible for the scheduling and the guarantee that a mobile agent will be delivered among Concordia Servers;
- *Directory Manager* provides naming service for applications and agents;
- *Event Manager or Inter-Agent Communication Manager* is responsible for registering, transmission and notification of events from one agent to another;
- *Agents Tool Library* is the group of tools and necessary classes that allows the development of Concordia mobile agents.

6 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 three phases of this process: configuration, release and reconfiguration of PVC in devices (hosts and switches) of ATM network.

6.1 Assumptions

The system is based on certain assumptions that result in a simpler process of development. These assumptions are necessary to isolate the main issues of this paper, trying to keep the applicability and extensibility of the proposed solution.

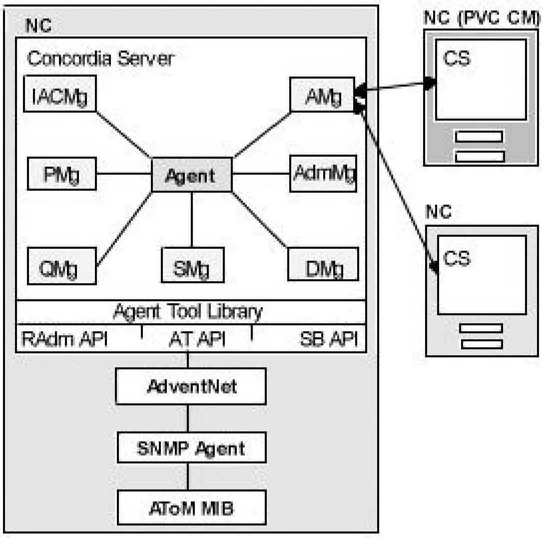
1. The functionality of the process is only related to the Configuration of point-to-point PVCs;
2. The class of QoS parameters can be freely configure, however, in order to simplify, the ‘best effort’ bandwitdth allocation parameter has to be used;
3. The user has the knowledge of the whole environment (hosts and switches) along the connection path, meaning then that the route is pre-defined and no decision about the routing should be made.

6.2 Implementation Architecture

The architecture of the system is constituted by the components defined below. All mobile agents in the prototype are implemented by using Concordia (Fig. 1).

The Concordia System 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 CR 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, responsible for the management of PVC configuration tasks of the devices, injects mobile agents into the ATM network. It specifies the group of switches along the PVC path, besides initializing the VPIs, VCIs, bandwidth etc.



NC: Network Component
IACMg: Inter-Agent Commnunication 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
AT API: Agent Transport API
PVC CM: PVC Configuration Manager

Fig. 1. Implementation Architecture

The AdventNet SNMP is a group of classes library written in Java to develop applications and applets for SNMP management networks. We adopted the AdvenNet v2c release 3.1 [10] since it supports the JDK 1.1 and higher ones. All other APIs and applications are projected for JDK 1.1. JDK 1.2 and more recent virtual machines. The package can be used to develop applications to manage SNMPv1 and SNMPv2 agents and contact systems of agents using any version of the SNMP protocols at the same time. All Concordia mobile agent interaction with the ATM MIB is done by importing classes of these components.

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.

The mapping of our Implementation Architecture in the Framework of the AMI Management Service can be found in [11].

7 PVCs Configuration Methodology

AtoM MIB has as its main focus the management of PVCs and the specification of its establishment, releasing and reconfiguration procedures. The methodology hereby proposed will describe each necessary step for the fulfillment of each phase mentioned above.

An important factor regarding the use of mobile agents in PVC configuration is to provide a uniform way for the ATM network operator to execute this operation. Therefore, it is no longer necessary to have the knowledge of the systems of various devices connected to a heterogeneous ATM network.

The end-to-end VC management using AToM MIB will be illustrated with an example of PVC configuration among the final systems 100.3.1.13, 100.3.1.4 and a intermediary system (switch 8285-100.3.1.2), involved in the project REMAV-FOR that belongs to LARCES-UECE. The VPI/VCI values, ports etc were used in the situation here analyzed [12].

Through the component PVC CM, the user starts the process by entry the PVC configuration data end-to-end, such as connection port and bandwidth of the switches that will be part of the virtual links. This way, a mobile agent is sent to the network to configure the PVC. Initially, with the requirements of the user, the mobile agent executes the task of configuration in the first host, then it migrates to the next switch. After configuring the switch, it travels to the following switch executing its configuration as well. These steps continue until the mobile agent reaches the final host and completes the task of configuration end-to-end. Consequently, it is a sequential procedure, since the mobile agent has to complete each task at each device before moving to the next one in order to complete a PVC. The VPIs/VCIs values are transmitted through the port of the configured device until the port of the next device.

When conditions of recoverable errors occur, the reconfiguration is done through a sequence of negotiations between mobile agents and devices. For example, recoverable failures occur when the VPI/VCI values selected by the PVC CM and already in use or when the bandwidth requested for the virtual link required is not available. When solving these kinds of errors, as well as when facing situations of negotiation of classes parameters and QoS, the mobile agent may need to return to the last device, by making intelligent decisions. The other kinds of failures can not be recovered. Thus they can not be negotiated.

We shall then present a detail of the steps of a PVC configuration mentioned above.

7.1 Establishment

The PVC establishment process consists of the following phases:

1. *Reserve appropriate VL* – the creation of a VL entry in the VL table (atmVpl/VclTable) by activating the row status atmVpl/VclRowStatus with CreateandWait. The PVC CM initiates to reserve VLs along the route by sending mobile agents to execute SNMP tasks to the ATM devices involved. If no errors occur, a row is created and VPC/VCI values are reserved on that port. The counters of VPCs/VCCs (atmInterfaceVpcs/Vccs) are automatically incremented. The interactions are shown on Table 1.

Destination	Interaction
Host 100.3.1.13	snmpSet(atmVclRowStatus.11.37.39=CreateAndWait)
Switch 100.3.1.2	snmpSet(atmVclRowStatus.11.37.39=CreateAndWait) snmpSet(atmVclRowStatus.3.25.27=CreateAndWait)
Host 100.3.1.14	snmpSet(atmVclRowStatus.3.25.27=CreateAndWait)

Table 1.

2. <u>*Characterize Traffic on the VL*</u> - The virtual link tables characterize the traffic to transmit and receive direction by pointing to the appropriate entries in the atmTrafficDescrParamTable. Multiple virtual links on the table atmVpl/VclTable can point to the same vector in the atmTrafficDescrParamTable.

The mobile agent characterizes the traffic parameters of all Virtual Links associated with the VC through the receive and transmit traffic index in the VL table to the atmTrafficDescrParamTable.

The VLs are activated by setting the row status (atmVclRowStatus) to **Active.** If errors do not occur, the reservation of resources to satisfy the traffic parameters values and the QoS Class for the VL will be completed.

3. <u>*Cross-Connect Virtual Links in the Intermediate Systems associating the VLs to the users application in the final systems*</u> – in the intermediate system (switch 100.3.1.2), the table atmCrossConnecttable should be used to cross the VLs connections. The tables atmVClTabale has an identifier column for this purpose (atmVclCrossConnectIdentifier). Different rows in the table atmVclTable that have the same identifier are cross-connected. This is achieved through cross-connect tables.

Before creating a row in the cross-connect table, a unique index must be obtained by using atmVp/VcCrossConnectIndexNext. A get-next will obtain a certain value. The VL cross-connect process consists of the following steps:

- creating a row in the cross-connect table;
- obtaining the value of the cross-connect index in the rows of the VL table;
- activating the row in the cross-connect table;
- turning on the traffic.

The necessary interactions in the intermediary system are listed in Table 2. At this point the traffic flow must actually be turned on.

Destination	Interaction

Switch 100.3.1.2	<u>1. Create a row in the atmVclCrossConnectTable</u>
	snmpSet(atmVcCrossRowStatus.3333.3.25.27.13.37.39=CreateAndWait);
	<u>2.Fill in the cross-connect index value in the corresponding VL table rows</u>
	snmpSet(atmVclCrossConnectIdentifier.13.37.39=3333);
	snmpSet(atmVclCrossConnectIdentifier.3.25.27=3333);
	<u>3.Activate the row in the cross-connect table</u>
	snmpSet(atmVclCrossConnectRowStatus.3333.3.25.27.13.37.39=Active);
	<u>4. Turn the traffic□□</u>
	snmpSet(atmVclCrossConnectAdminStatus.3333.3.25.27.13.37.39=Up);

Table 2.

Finnaly the traffic in the computers is activated by issuing the values **Up** to the atmVclAdminStatus row of its tables atmVpl/Vcltable.

All the steps above can be shortened by issuing the **CreateAndGo** value to the row status objects (atmVclRowStatus). This way, it is not possible to obtain a detailed error analyses. Thus the step-by-step process is recommended.

7.2 VL Release

The VL release consists of two phases:

1. Release the cross-connects in the IS – to release the VL, all cross-connects and associated VLs must be released by associating **Destroy** to the row status in the table atmVc1RowStatus. This will liberate the atmVc1CrossConnectRowStatus value for future use by atmVCCrossConnectIndexNext and the atmVc1CrossConnectIdentifier will be removed from the associated VL.

2. Release the Virtual Links – to restore the associated VLs to the VC, each atmVc1RowStatus entry of the atmVc1Table of each device must be destroyed.

Upon these action, the SNMP agents will release the associated VL resources and decrement atmInterfaceVccs. It is recommended to release the cross-connects before destroying the VLs individually. Otherwise, if the VL is released first, in many implementations, it can be interpreted as a request to change configuration.

3. Release the Traffic Descriptors – to release the traffic parameters associated with transmit and receive directions of the virtual links, the rows of the traffic descriptor table (atmTrafficDescrParamTable) pointed to by the virtual links must be deleted.

7.3 VL Reconfiguration

The main reconfiguration applications consist in the following changes:

1. Traffic and/or QoS Parameter value changes – In this case, an additional capacity of the SNMP agent is not required. The mobile agent takes down the current VC and defines new virtual links with the desired parameter and creates a new VC by following the rules described above.

2. Topology Changes – a topology change, opposed to the reconfiguration described above, requires additional capacity of the SNMP agent, including the hardware/software support.

8 Conclusion

This paper presented a PVC configuration management methodology with the use of mobile agents developed in a Concordia environment. Thus, the PVC configuration manager has an overview of all devices belonged to the network. The user does not have to worry about the system of each switch and is able to delegate the responsibility of configuration to the mobile agent. Although the specific MIB information of each maker is stored in different systems, its access is possible by using the AdventNet. The mobile agent automated the PVC configuration tasks without the need of interventions by the users in the decisions.

The methodology presented is a sequence due to the natural characteristics of the PVC configuration procedure. Like [13] some studies on the mobile agent launching are being done based on parallel methodology since the time spent to configure the nodes tend to be less than if a serial methodology was used.

Although a number of assumptions were made, this paper focus on a great number of relevant aspects to the project and implementation of real architectures of mobile agent systems. The results here obtained are a great indication that operation with mobile agents has significant impact on the performance of management network application.

References

[1] http://www.omg.org - "Corba Overview", OMG.
[2] "Instant CORBA "Roberto Orfali, Dan Harkey, Jeri Edwards 1997 John Wiley & Sons, Inc. ISBN 0-471-18333-4.
[3] http://www.omg.org/about/wicorba.htm "what is CORBA" OMG.
[4] STALLINGS, W. SNMP, SNMPv2, SNMPv3, and RMON1 and 2, Addison Wesley, Third Edition, 1998
[5] IETF WWW Pages URL= http://www.ietf.cnri.reston.va.us/home.html
[6] ATM Forum WWW Pages URL= http://www.atmforum.com
[7] SPRENKELS, R.A.M., "Management of ATM Networks", M.Sc-thesis, June 1996.
[8] AHMED, M., e TESINK, T., Internet Engineering Task Force, RFC 1695, "Definitions of Manged Objects for ATM Management, Version 8.0 using SMIv2," Agost 1992.
[9] Concordia White Paper: Mobile Agents Computing URL=http://www.meitca.com/HSL/Projects/Concordia/MobileAgentsWhitePaper.html
[10] AdventNet v2c Release 3.1 URL= http://www.adventnet.com
[11] CARDOSO, A.R., e JR CELESTINO, J., Implementing Mobile Applications Based on AMI Architecture for Network Management, IFIP/IEEE International Conference on Management of Multimedia Networks and Services 2001 Chicago Il, USA, october 2001 (Submitted)
[12] PAN, H., SNMP-Based ATM Network Management, Artech House, Hardcover, September 1998.
[13] BOYER, J., PAGUREK, B., WHITE, T., Methodologies for PVC Configuration in ATM Heterogeneous Environment Using Intelligent Mobile Agents, First International Workshop on Mobile Agents for Telecommunication Applications MATA 99, Ottawa, Canada, october 1999.