Service Provisioning and Management in Telecom Environments Using Active Network Technology

Fábio Luciano Verdi and Edmundo R. M. Madeira

Institute of Computing, University of Campinas (UNICAMP), Campinas-SP, Brazil
Telephone: +55 (19)-3788-5862
Fax: +55 (19)-3788-5847
{verdi, edmundo}@ic.unicamp.br

Abstract. The advent of Telecom over the last years brought up many challenges to service providers. The difficulty to offer services and, at the same time, to perform their management requires an integrated solution for both, service provisioning and management. In this paper, we outline an infrastructure to perform such tasks. This infrastructure allows to create Virtual Active Networks (VANs) and install services in Telecom environments based on the Active Networking technology. Besides service provisioning, the infrastructure performs some management functions taking into account mobile services. The management system is based on a mobile agent platform and considers the management of VANs, where services can migrate from a VAN to another. Some aspects about the implementation of a prototype we developed to test our approach are presented. In this prototype we focus on three specific management areas: accounting, performance monitoring and configuration.

1 Introduction

Service management in Active Networks (ANs) is currently a potential topic of research. Mobile Agent Technology (MAT) is a general name for facilities that support the transmission of code, as well as data, over a computer network. In this context, the AN technology is particularly attractive to be used in Telecom environments, due to its facility to send data and code to specific locations in the network. In the AN technology, the packets, also called *active packets* or *capsules*, can carry programs to be executed on routers and possibly change their state.

With AN technology, the customization of services can be achieved by partitioning the network into *Virtual Active Networks* (VANs), which customers can lease from service providers. A VAN can be described as a graph of virtual active nodes connected by virtual links where active packets can travel within a VAN or among different VANs [2]. The VAN concept is a very useful solution to isolate groups of customers and offer specific services to each customer's domain.

A possible solution to support the interaction between providers and customers is to develop a framework, which allows a customer to install, configure

E. Horlait (Ed.): MATA 2003, LNCS 2881, pp. 32–41, 2003.

and run active services on his own VAN. These services can be static, or they can be moved from node to node within a VAN or among VANs. Information about accounting, performance and configuration enables the manager (human) to know the behavior of the managed environment to detect problems and take actions on them. In this paper, we describe a framework for service provisioning and facilities for policy-based management of these services. The services offered by the infrastructure are VAN creation and service installation, both implemented using the Active Node Transfer System (ANTS) [7].

Many approaches taken today have focused on AN technology. In this context, an architecture for network management that offers active services is presented in [3]. A model to manage mobile services in VANs is described in [5]. An approach for service management in Telecom environments is presented in [2]. The Mobile Agent paradigm and its use in the context of telecommunications is described in [4]. Taking into account the features from these approaches, our framework has incorporated some important aspects for Telecom environments such as: policy-based management, VANs, mobile services and mobile agent-based management. In contrast to [1], in which services are based on mobile agents, in our model the services are moved from a VAN to another using the active networking infrastructure. We have tested our approach on the Virtual Private Active Network (VPAN) service which is fully explained in [6]. An example of using such a service is when a group of researchers located in different companies needs to work together in a project, sharing resources and/or sending and receiving applications to be tested in different domains (e.g. integrated development of software).

The paper is organized as follows. Next section outlines our infrastructure and its components, and it illustrates four possible management scenarios. Section 3 describes some aspects about implementation and Section 4 presents the conclusion.

2 The Framework for Service Installation and Management

The framework we developed includes components to support VAN creation, service installation and management in virtual active Telecom networks. In this section we present these components, the defined policies, the configuration actions and, additionally, we focus on four possible Telecom scenarios which our model is able to handle. The management system we describe in this section is to manage a single domain. In [6] we address the extensions needed to support multi-domain management.

2.1 Policies

We have defined two possible solutions that can be realized when a client requires a service. The first solution considers that whenever a client needs a service, a new copy of this service will be created and sent to the client. The second alternative considers that whenever a client requires for a service, possibly a copy of this service from a node will migrate to attend the customer requirement. Both solutions are unfeasible whether their consequences are considered. To install a new service copy for each customer will flood the network with services. To migrate a service from a node to another to attend customer requirements will result in many migrations. Thus, we have tried an intermediate solution defining some policies to minimize both, the number of service copies and the number of migrations.

We created six representative policies:

1. There are three kinds of service: internal services which can only migrate in the same VAN, external services which can migrate among VANs, and VPAN services which can migrate among domains that belong to the same VPAN. The latter is explained in [6];
2. A service S has a limited number of copies in the network (**L1**): S_1,S_2,...,S_n, where n$\leq$**L1**;
3. A service S has a maximum bound of times to migrate in a period of time (**M1**). Each copy of S, S_i, has to follow this threshold;
4. The SP (Service Provider) can install a new service copy requested by a user until a threshold of **L2** (L2$\leq$L1).
5. The Least Recently Used Service (LRUS) will be the candidate copy to migrate to the destination host when a customer requires a new service. This happens if the threshold L2 was already reached;
6. A service copy may be removed if it does not migrate a minimum bound of times in a period of time.

Next, we present the cases that can happen considering these defined policies.

Analysis of Cases: Policies 2, 3, 4 and 5 defined above were created to implement our intermediate solution. There are four possible cases to be considered when a client requests a service. Let $\mathbf{S}_i$ to be the ith copy of service S, **C(S)** to be the number of copies of S in the network and $\mathbf{M(S}_i\mathbf{)}$ the number of migrations performed by S_i:

1. **C(S) $<$ L2**: in this case the number of copies of S is less than L2 and, therefore, the SP can install a new copy of S. This minimizes the number of migrations;
2. **C(S) $\geq$ L2 and C(S) $\leq$ L1 and there is the LRUS, $\mathbf{S}_{lrus}$, with $\mathbf{M(S}_{lrus}\mathbf{)}$ $<$ M1**: in this case, the threshold L2 was reached but the maximum number of migrations of S_{lrus} not and, therefore, S_{lrus} will migrate to attend the customer requirement. This minimizes the number of copies;
3. **C(S) $\geq$ L2 and C(S) $<$ L1 and the LRUS, $\mathbf{S}_{lrus}$, with $\mathbf{M(S}_{lrus}\mathbf{)}$=M1**: in this case, the maximum number of migrations of S_{lrus} was reached, so this service will not migrate during a period of time, but it is possible to create a new service copy since C(S) $<$ L1. During the interval from L2 to L1 there is a balance between creating new copies and migrating services;

4. **C(S) = L1 and the LRUS, S_{lrus}, with M(S_{lrus}) = M1**: in this case, the maximum number of migrations of S_{lrus} and the maximum number of copies of S were both reached. The service will not be installed.

2.2 Configuration Management

We defined three types of possible actions to apply: create a new service copy (1), move a service for load balancing (2) and delete a service from the environment (3). The first one is responsible for creating a new service copy when cases 1 and 3 (as explained in Section 2.1) are considered. The second allows the manager to move services from a node to another in order to balance the load. The last one is useful to remove services which were created in order to attend a period of demand and no longer are being used.

2.3 Components of the Infrastructure

The infrastructure is composed of a Management Remote Application (MRA), a Management Center (MC), a Service Provider (SP), a Global Naming Service (GNS) and the Distributed Management Agents (Local Manager - LM and Managed Active Element Agent - MAEA) as described below (Fig. 1).

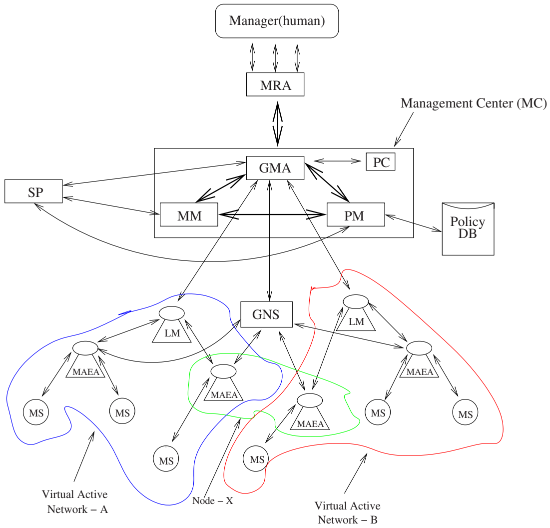

Fig. 1. Components of the Infrastructure and their relationship.

– **Management Remote Application - MRA**: This application presents to the manager (human) the interface with the methods related to the management questions (accounting, performance and configuration).

- **Service Provider - SP**: It is responsible for offering services to the customers. Clients send capsules to the SP to create a VAN and its services or to add a new service to their VANs.
- **Policy Manager - PM**: This component is responsible for controlling, insertion, updating and removal of a policy in the database.
- **Migration Manager - MM**: This component sends accounting management agents to the nodes and controls each service migration over the network applying the specific policies.
- **Global Manager Agent - GMA**: It is the centralized static manager of the model. There is only a GMA in a domain. This component is responsible for analyzing accounting, performance and configuration data on all VANs and their services.
- **Local Manager - LM**: There is an LM per VAN. It is responsible for the management of its VAN and collects data of each *Managed Active Element Agent* (MAEA, see below) located in the hosts of the VAN.
- **Managed Active Element Agent - MAEA**: It is responsible for managing one or more services in a host of a VAN, being the lowest level of the management. There is an MAEA per host per VAN. This agent is an interface between the ***Managed Service*** (MS) and the management system.
- **Global Naming Service - GNS**: This component receives the service location and the service identifier and creates a reference that indicates where the service is located.

Figure 1 shows two different VANs, A and B. We can see that node X belongs to both, but the model separates each one for service management.

2.4 Service Provisioning and Management Scenarios

The next four scenarios are associated with the customer's *Active Application* (AA), that is a customer's program to use the available environment. The first scenario describes in details the VAN creation and the mobile service installation. The second one presents a service installation following case 1 (see Section 2.1), i.e., without migration. The third scenario presents a mobile service migration between two VANs to satisfy a customer requirement. This scenario follows case 2 from Section 2.1 in which the threshold L2 was reached. The last scenario follows case 3 from Section 2.1 and it represents the situation in which the LRUS cannot migrate but a new service copy can be created. These four representative scenarios allow to demonstrate how our framework can be applied to offer VAN services and manage a typical Telecom environment.

1. **Creating a new VAN specifying what hosts and what services are required**.
   In this scenario the following sequence is needed, as shown in Fig. 2.
   Customer sends a capsule to the SP informing what hosts and what services each host will have (a); the SP sends the required services to the set of hosts (b); the SP notifies the MM and GMA about the new VAN (c); the

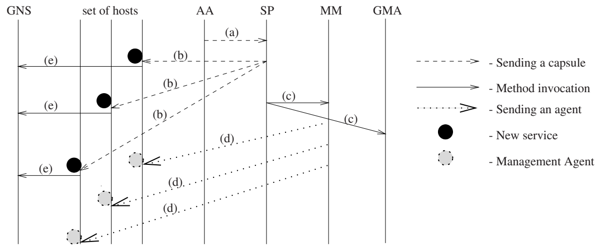

Fig. 2. Customer creating a new VAN.

MM sends accounting management agents to the hosts and creates the VAN manager (LM) (d); and finally services register themselves in the GNS (e).

Next, we present the three possible scenarios to add a new service copy to the VAN. Figure 3 shows the first three steps (a-c) which are necessary to all of the three scenarios. These three steps are: customer sends a capsule to SP indicating what service is required and what destination host this service copy will be sent to (a); SP interacts with the MM in order to apply the policies on the required service (b); and MM gets the service policies from the PM to analyze them (c).

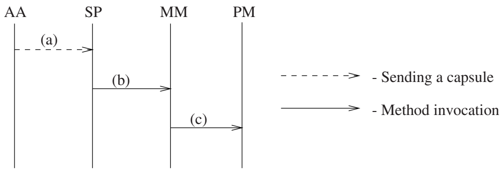

Fig. 3. Customer requiring a new service.

After performing these three first steps, one of the following three scenarios can be executed based on the results obtained by the SP.

2. **Adding a service to the VAN following case 1.**

   In this case, the threshold L2 has not been reached, so the SP installs a new service copy in the VAN to attend the client request. In addition to the first three steps from Fig. 3, the following steps are necessary (Fig. 4):

   SP sends a capsule to the destination host to create a new copy of the required service (d); SP notifies the GMA about the provisioning of a new service copy (e); the capsule arrives in the destination node, instantiates the new service copy, and registers it in the GNS (f); the service notifies its local MAEA that it has arrived (g); the capsule is forwarded to the new AA to notify that the required service has migrated. The AA can finally use the service (h).
3. **Adding a service to the VAN following case 2.**

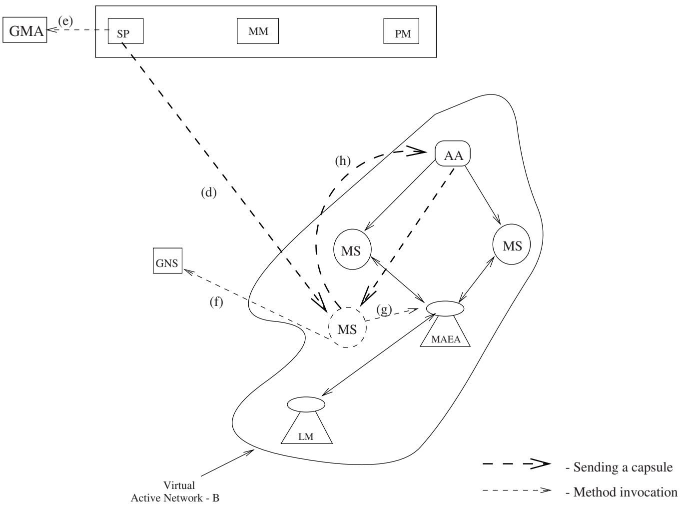

Fig. 4. SP installing a new service (case 1).

In this scenario it is considered that the limit L2 from policy 4 was achieved but the number of migrations not. If the required service is external, the LRUS can belong to another VAN and, then, the service must migrate between VANs. In addition to the three steps from Fig. 3, the following steps are necessary (Fig. 5):

SP gets the LRUS in the network. In this search, all the information about the LRUS is also obtained (d); SP interacts with the MM in order to verify whether the migration is possible or not (e); MM gets the service policies from the PM to analyze them (f); MM returns back the results to the SP (g); SP sends a capsule to AA of the VAN where the LRUS is located. This is necessary to notify the AA about the LRUS migration (h); SP notifies the GMA about the migration (i); the capsule is forwarded to the host where the service is located at this moment (j); the service unregisters itself from the GNS and notifies its MAEA that it is migrating (l); the source MAEA notifies the destination MAEA that a new service is arriving (m); the capsule removes the service from the local node and migrates it to the destination node (n).

To conclude the service installation, steps *f, g* and *h* from scenario 2 (Fig. 4) are executed.

4. **Adding a service to the VAN following case 3.**

   In this case, the LRUS cannot migrate during a period of time, but a new service copy can be created and sent to the client. In addition to the steps from Fig. 3, steps *d, e, f* and *g* from scenario 3 (Fig. 5) are performed to find the LRUS and apply the policies on it. After the SP realizes that the migration is not possible but a new service copy can be created, steps *d, e,*

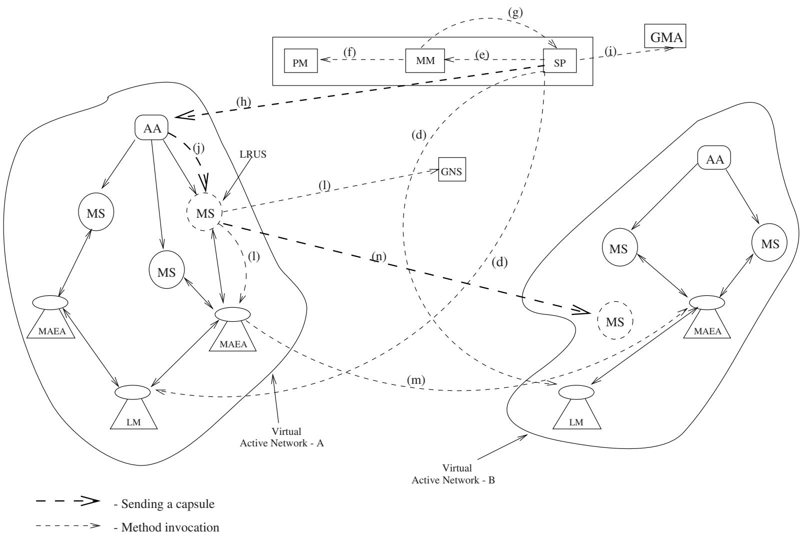

Fig. 5. Migrating the LRUS to attend the customer requirement (case 2).

f, *g* and *h* from scenario 2 (Fig. 4) will be performed to conclude the service installation.

In scenario 3, each LM is responsible for finding the LRUS in its VAN and returns back the result to the SP. If the required service is internal (see policy 1), the same steps are performed, but in step *d*, SP obtains the LRUS only for the local VAN. For all other steps, the destination VAN is the local one.

3 Some Issues about the Implementation

To create the environment with VANs and services migrating among them, we used the ANTS toolkit. We have mainly used the following classes from ANTS: **Application**: in order to allow the customer to install and use services in the active nodes; **Capsule**: to send and receive code to and from the nodes; and **Node**: this class represents the execution environment of a VAN in a host. The active environment and the management system have been developed using Java 1.2. The services are implemented as Java objects. The mobile agent platform is the Grasshopper 2.1 [8], and the active nodes of each VAN are running on Sun Workstations executing SunOS 5.5. A service has a single name and is registered in the GNS. The way the service name is created is explained in [5].

The prototype allows to get management information in different levels: per service, per host, per VAN and per domain. Figure 6 shows a general view of the environment. It is possible to know, among other information, the throughput per host and the use of cpu and memory in each host.

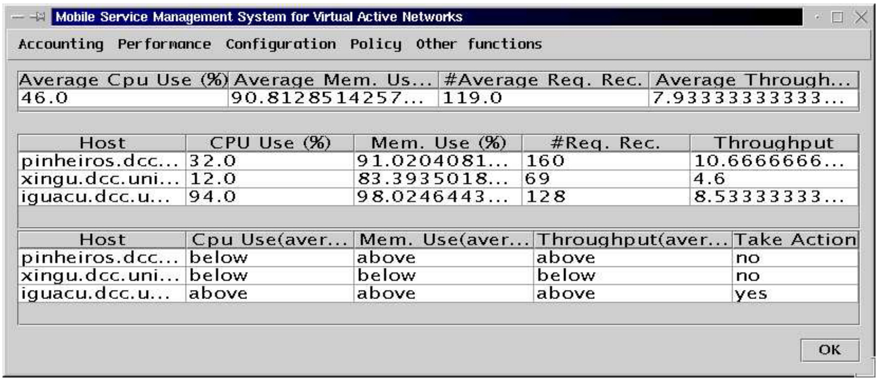

Fig. 6. Management Remote Application in a general view.

It is also possible to keep track of service migrations. Figure 7 depicts a service migration and some accounting information in each host. In this management operation, the service is named *VAN11.1.1.2CallForwardingG*. We can see that the service has migrated between VAN1 and VAN2 in order to attend the customer roaming. Initially, it is in the host 1.1.1.5 belonging to the VAN2. After some migrations, it is in the host 1.1.1.2 belonging to the VAN1. Also, below each host, we can see some information like the number of received requests and residence time.

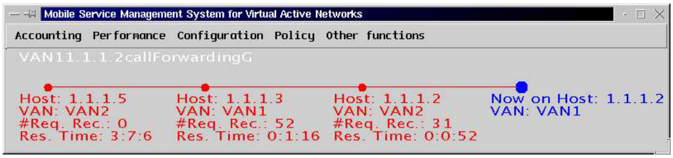

Fig. 7. Service migration.

Figure 8 shows the most requested service in VAN2. We can see, among other information, the service name, the host where the service is located and the number of requests received by this service.

4 Conclusion

In this paper we presented a framework for service provisioning, such as VAN creation and service installation in Telecom environments based on VANs. The framework also performs the policy-based management of these services and their migration. The environment has a Service Provider, and a Policy Manager

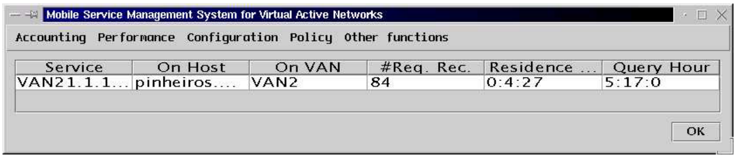

Fig. 8. The VAN2 most requested service.

responsible for controlling and processing policies. The proposed management model is based on a mobile agent platform and considers three management functional areas: accounting, performance and configuration. This model is flexible in the sense of management questions can be answered from different levels: per service, per host, per VAN, and per domain.

We create some scenarios to simulate a Telecom environment: exploiting the VAN creation, installing mobile services, and managing a service migration. The policies we defined gave us a good comprehension on how our intermediate solution is useful to minimize the number of service copies and the number of migrations. The implementation has showed that the proposed framework can handle Telecom environments with mobile services and VANs successfully, opening up a potential further study.

Acknowledgements

The authors would like to thank CNPq and CAPES for their support.

References

1. M. Breugst and T. Magedanz. Mobile Agents - Enabling Technology for Active Intelligent Network Implementation. *IEEE Network,* pp. 53-60, May/June 1998.
2. M. Brunner, B. Plattner, and R. Stadler. Service Creation and Management in Active Telecom Networks. *Communications of the ACM,* pp. 55-61, March 2001.
3. I. W. Marshall, H. Gharib, J. Hardwicke, and C. Roadknight. A Novel Architecture for Active Service Management. *Seventh IFIP/IEEE International Symposium on Integrated Network Management,* Seattle, Washington, USA, May 2001.
4. V. A. Pham and A. Karmouch. Mobile Software Agents: An Overview. *IEEE Communications Magazine,* pp. 26-37, July 1998.
5. F. L. Verdi and E. R. M. Madeira. A Mobile Agent-based Model for Service Management in Virtual Active Networks. *IFIP/IEEE 12th International Workshop on Distributed Systems: Operations & Management - DSOM'01,* Nancy, France, pp. 101-112, October 2001.
6. F. L. Verdi and E. R. M. Madeira. Service Provisioning and Management in Virtual Private Active Networks. *IEEE 9th International Workshop on Future Trends of Distributed Computing Systems - FTDCS'03,* San Juan, Puerto Rico, pp. 205-211, May 2003.
7. D. J. Wetherall, J. V. Guttag, and D. L. Tennenhouse. ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols. In *IEEE OPENARCH'98*, San Francisco, CA, April 1998.
8. IKV++ GmbH - Grasshopper: http://www.grasshopper.de.