Supporting Nomadic Users within Virtual Private Networks

Stamatis Karnouskos
German National Research Center for Information Technology
Research Institute for Open Communication Systems (GMD-FOKUS)
Kaiserin-Augusta-Allee 31, D-10589 Berlin, Germany
email: karnouskos@fokus.gmd.de

***Abstract* - Virtual Private Networks (VPNs) are communication environments for groups deployed within a public network but are actually not taking care of mobility. Nomadic VPN users have special requirements relating to resource adaptation and customization. This paper discusses how mobile agent technology applied to DPE-based VPN provides seamless access, service capabilities to nomadic users within the visiting network environment. Taking into account the structuring mechanisms enabled by standard mobile-agents platforms, such as regions, agencies grouped within regions, and places belonging to agencies, we have to apply these structural principles to our target mobile communications environment. Currently service providers don't have access to nodes' control environments, algorithms and states. Within programmable DPE based VPNs, agents traverse the network and configure the nodes in order to achieve the best of the network's behavior during a particular task/application data flow between VPN users.**

I. Technology Context

Technology integration is the main challenge we address when considering Nomadic Virtual Private Network. Nomadic users may be supported either separately or in small groups. The context in which this group operates is considered as the object of adaptation in order to satisfy nomadicity. We investigate how the combination of software agent technology and programmable networks could help this adaptation at different levels.

A. Virtual Private Networks (VPNs)

A VPN is a communications environment in which access is controlled to permit peer connections only within a defined community of interest and is constructed through some form of partitioning of a common underlying communications medium, where this underlying communications medium provides services to the network on a non-exclusive basis. Such kind of networks are deployed within a public network and aim at providing a private working environment to its users while also taking advantage of the efficiencies of the underlying infrastructure.

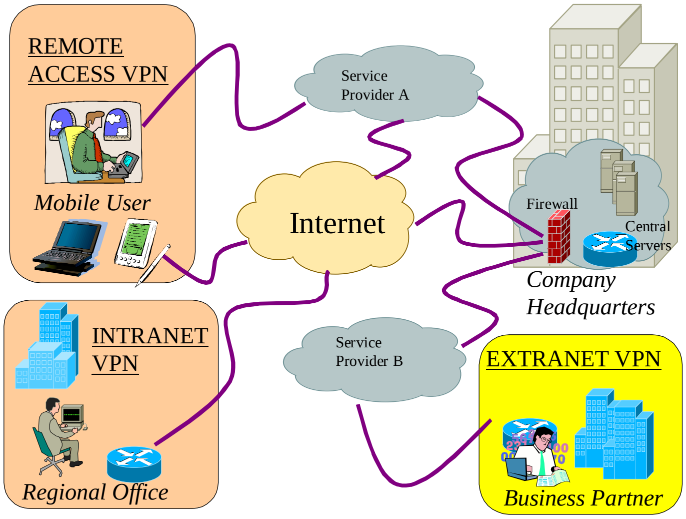

Fig. 1. VPN Systems

As also depicted in Fig.1, there are various types of VPNs depending on the functions they perform as well as on the different methods for constructing them e.g. extranet VPNs, intranet VPNs, access based VPNs, hardware-based VPNs, firewall-based VPNs, Software-based VPNs etc.

The bottom idea is that VPNs provide an abstraction of a network infrastructure that is used by groups or enterprises to cover their needs. Security features differ from product to product, but usually include encryption, strong authentication of remote users or hosts, and mechanisms for hiding or masking information about the private network topology from potential attackers on the public network. VPNs are designed mostly with static users in mind and little has been done to easy integrate mobile users or to provide mobile user support after their deployment.

B. Nomadic Users

Nomadic users are wanderers, people on the move from place to place. The goal is to make information services and applications ubiquitous and flexibly available for such individuals as well as to small groups of them. The problem is that the need for and the availability of information and communication services vary from place to place and from time to time. Key requirements are the a) rapid service adaptation and customization and b) security. We are mostly interested two categories in:

a) *An individual or a group of individuals moving together*. The aim here is to maintain the local context as the group as a whole moves. Services provided to the group should be the same even though the group or the individual (group with one member) is away from the home environment. A military squad in a battlefield falls within this category. Here the connections between the group members may be intact and only the underlying infrastructure changes. Thus the services provided to the group have to be adopted and this should be done without any significant changes to the upper levels.

b) *A distributed context with autonomously moving members*. The aim here is to keep a virtual community and its context intact by rapidly adapting to the new environmental parameters that are generated by the move of its members. A multi-conference between mobile users falls within this category. Here the matters get more complicated as two parameters change i) the underlying infrastructure ii) the connections among the members of the group.

We will try to address both areas specified above and we will propose a flexible way to deal with the majority of problems that arise in these contexts.

C. Active/Programmable Networking

Active and programmable networking [8] is a quite new technology that aims at transforming current networks from passive data carriers to active, dynamically configurable infrastructure that not only transports data but also performs computation on those data. This is accomplished by opening-up the network node interfaces to third party entities. Active networking allows applications to customize both control and data where programmable networks allow the programming of signaling and control functions but with fixed data transfer functions.

In our case, as nomadic users can be spread to various locations with heterogeneous infrastructures, flexibility as well the programmability of the network are much needed in order to fully support the new requirements that come into the scene.

D. Mobile Agents

Agents [2] are software components that act alone or in communities on behalf of an entity and are delegated to perform tasks under some constrains or action plans. One key characteristic of agents is mobility, which allows them to transport themselves from node to node and continue their execution. Mobile agent technology has established itself as an improvement of today’s distributed systems due to its benefits such as dynamic, on demand provision and distribution of services, reduction of network traffic and reduction of network dependencies, fault tolerance etc. Mobile agents can migrate along with nomadic users, adapt local and remote resources dynamically and generally manage and mediate all nomadic user’s requirements.

II. THE REQUIREMENTS

The requirements of nomadic users and the infrastructure that supports them can be viewed from a number of perspectives: The portability perspective that involves the user, the home environment and the serving network, and VPN’s perspective that deals with group support.

A. Portability Requirements:

User requirements: The user wants to freely move in heterogeneous environments and be able to customize the services offered to him, to personalize the user interfaces based on terminal’s capabilities, have ubiquitous access to all services offered to him independent of his location, the ability to modify his profile and service activation/deactivation from any location, to be able to discover the additional services in the new environment and have all of the above in an optimized and cost-effective way.

Home environment requirements: The home environment wants to provide a high customization of its services to the users it hosts. It wants also to provide an easy way to make these services available even when the users roam in third party networks. Access controlled access to the services offered should be fine-grained based on the user credentials or groups or even the foreign network or location of the user.

Serving network requirements: The serving network may offer to its temporary visitors access to some of its capabilities or merely provide a connection to the home environment. It has to provide the visitor with transparent access to the services he subscribed in his home environment and additionally offer him new services not available in the home location. Billing and management of visitor users is also a challenge.

B. VPN’s Requirements

Admission Control: A policy specifies who can access the network and its services. By providing the services selectively to restricted members, customization could be applied. An enforcement engine enforces the policy between the members whose privileges and obligations vary.

Information Path Selection: The topology, as well as the path that information follows within a network, need to be dynamically adjustable. This can be due to security, customization, effectiveness, QoS requirements or other reasons. For instance, we might require that our flow follows a specific route via trusted hosts or dynamically adjust to available bandwidth and select always the highest bandwidth connections links.

Resource Management: VPNs are statically or dynamically assigned some resources (communication bandwidth, memory allocation, disk space allocation etc) that are used by its group members. The VPN is responsible for the reservation and usage of its resources by the users. This task gets complicated as the users are mobile and the VPN is spited between the home network and various foreign networks

Security/Privacy: Security and privacy are fundamental needs of an enterprise. We need authorization and identification schemes in order to screen and serve requests.

Effectiveness/Performance: The VPN must operate at highest performance and take advantage of the underlying network resources and services in their optimized form.

Flexibility/Adaptivity: The VPN has to react to environmental changes (key issue in nomadicity) and adapt its behavior in order to avoid malfunctions and achieve flexibility and effectiveness.

III. Technology Integration

To flexibly integrate the above-mentioned technologies, we consider some guiding principles. Each service is provided to the user by a single or co-operative set of agents. The user agent makes all the communication in order to connect the user with his services in the new visiting environment and is generally responsible for reacting on behalf of the user for whatever environmental changes occur. The user is consulted only for critical decisions that cannot be handled by the agents themselves. The aim is to make all changes transparently, and with minimal human intervention. We describe first the agent system and its basic characteristics and then we embed it to a programmable node. This is our infrastructure and later we will demonstrate with a scenario its usage.

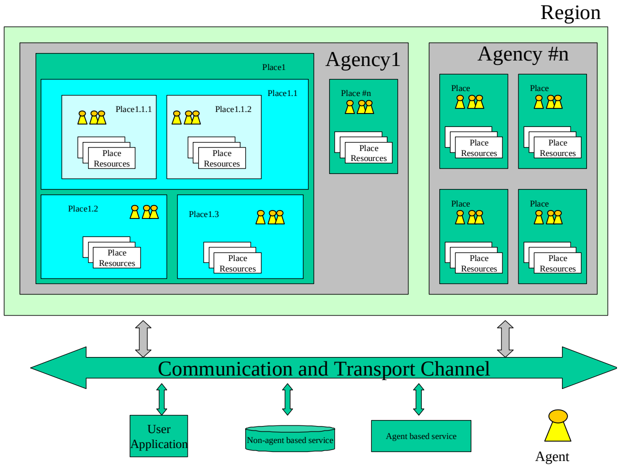

Fig. 2. Agent EE Architecure

A. The Agent System

An agent system [12] (Fig. 2) is a platform that can create, interpret, execute, transfer and terminate agents. More than one agent system can co-exist in an operating system. Agent systems consist mainly of places. A place is a context within an agent system in which an agent is executed. This context can provide services/functions such as access to local resources etc. A place is associated with a location which consists of a place name and the address of the agent system within which the place resides. Places can contain other places (nesting places). All places follow the parent-child paradigm of Unix processes in the way that each child is assigned/makes use of its parent’s resources. Also its policy is an extension/customization of its parent's policy etc. Places are a) dynamically assigned to agents as they enter the node based on some criteria e.g. all agents coming from a specific user or location or agents belonging to a specific policy scheme or b) statically assigned per entity (e.g. user, enterprise etc). Here static resources are given to the place (after agreement with the node provider) and the local resource manager manages them. In this way it is possible for an enterprise to setup a network of places in various nodes, creating an execution environment oriented Virtual Private Network [7]. This offers several advantages e.g. secure communication between company-trusted places. A policy manager and a resource manager mediate access to resources assigned to each place.

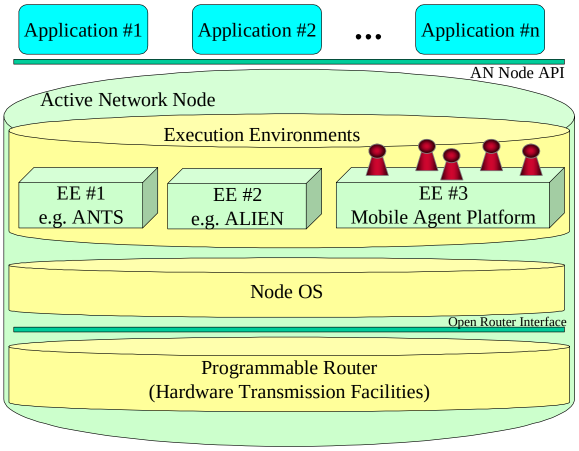

Fig. 3. Agent-based Active Node

The existence of different Execution Environments (EEs) for agents that have the same owner/characteristics serves the need to avoid unwanted interactions. Isolation done by EEs is similar to the sandbox idea that exists in Java. Since in each place agents with common characteristics (e.g. of the same owner) are gathered the possibility of attacking each other is lower as usual. Beyond having unique IDs, also hold their own public/private keys for authentication and digital signing purposes.

B. The Agent-Based Active Node

The agent system described above is embedded in the active node as shown in the architecture below in Fig. 3. We can distinguish in the architecture above the following parts and entities:

A Programmable Router: The router is accessed via an interface for dynamic programming of its resources. The open node interface represents the abstraction of the router resources ranging from computational resources (CPU, memory etc) to packet forwarding resources (bandwidth, buffer, etc). The APIs are standardized by the IEEE P1520 project [1]. By opening up the router resources, more advanced functionality can be built in the upper levels.

The Node OS: this is the operating system running on each node in an active network. The NodeOS provides the basic functionality from which the EEs built the abstractions presented to the active applications. The architecture of the NodeOS and its functionality is outlined in detail by the AN Node OS Working Group [9].

Execution Environments: which are on top of the NodeOS, making use of its services. The functionality of the active network node is divided among the Node Operating System, the Execution Environments (EEs) and the active applications. The architecture allows multiple EEs of various providers to co-exist and be present on a single active node. Each EE (e.g ANTS [10], ALIEN [11], Agent EE) exports a programming interface or virtual machine that can be programmed or controlled by third party code. The NodeOS manages the resources of the node. One of the EEs is the Mobile Agent EE where agents execute when they visit the node.

Cooperating Agents: They reside in the Agent-specific EEs and via the facilities offered to them program the node. These can be either mobile agents (e.g. visiting agent) or even stationary intelligent ones that reside permanently in the EE implementing various services. The agent can either be generated at a place locally (e.g. out of a pool of ready-programmed objects) or it can just carry on with an execution it suspended in another node.

IV. FLEXIBLE VPNS WITH NOMADIC USER SUPPORT

The main idea adopted to support the nomadic user's requirements is to apply a structure of the Agent DPE. Each user can be considered as acting within his own place (as defined in agent terminology). This place (that is under the total control of the user) hosts one or more cooperating agents that keep track of the user's needs and current status. Furthermore these agents are responsible for mediating the services that exist among the users. When that specific user moves from one network point to another, the agents are responsible for providing optimal adaptation to the new environment and reconnect/reconfigure the services that the user needs in order to provide the same (not only in look and feel but also in functionality) working environment as before. The whole process should be transparent to the end users.

A VPN with nomadic users constitutes a graph with changing nodes (due to mobility requirements). The challenge is to re-assign the connections between the nodes of the graph in order to provide the same services in a higher level despite of the fact that the underlying infrastructure continuously changes as the nomadic users move.

We believe that agent technology in combination with the active/programmable networks are the right step to this direction. Agents can also be intelligent, which means that they can adapt easier to non-deterministic environmental changes, learn while they are active and act proactively in order to satisfy their internal goals in a heterogeneous infrastructure. We will demonstrate with a scenario the two-folded objective of VPNs with nomadic user support and how it is achieved. Firstly we will show how agents can be used to deploy the initial VPN among the end users. Subsequently we will consider two users as nomads and we will explore the infrastructure adaptation while these users change location and terminals.

A. Initial VPN provisioning

Our objective is to dynamically and flexibly provision VPNs, allowing the deployment of a VPN in minimal time. Agents here take the roles of the customer, the service provider and the network provider. The agents are signed with their users' private keys, a proof that their users delegated them with the specific tasks. There are multiple network and service providers in the infrastructure on top of which we want to build our VPN. We assume that the group agent (GA) interacts with the VPN service provider agent (SPA) and with the network provider agent (NPA). The SPA has to negotiate with all UAs and NPAs. If a specific service provided by a service provider spawns multiple domains then the service provider makes the negotiations concerning the service with those domains. The stages to follow are:

Common Requirement Definition Stage: In this stage the user agents (UAs) negotiate and come up with a common set of requirements for the underlying infrastructure and services that is desired. The UAs assign as responsible for the further negotiations a GA and all sign the common set of requirements.

VPN Network Negotiation Stage: In this stage the GA negotiates with NPAs the possible topology of the VPN and the requirements in the networking infrastructure. This is done if users have a specific requirement on network e.g. spotted nodes via which their communication should pass.

VPN Service Negotiation Stage: In this stage the GA has multiple network topologies that fulfill his requirements. Subsequently he negotiates with SPAs in order to see which of the possible topologies support the desired services. The service provider can also offer alternative solutions concerning the network topology etc. At the end of this stage the GA has some network topologies that support the services he requires and a set of service level agreements (SLAs).

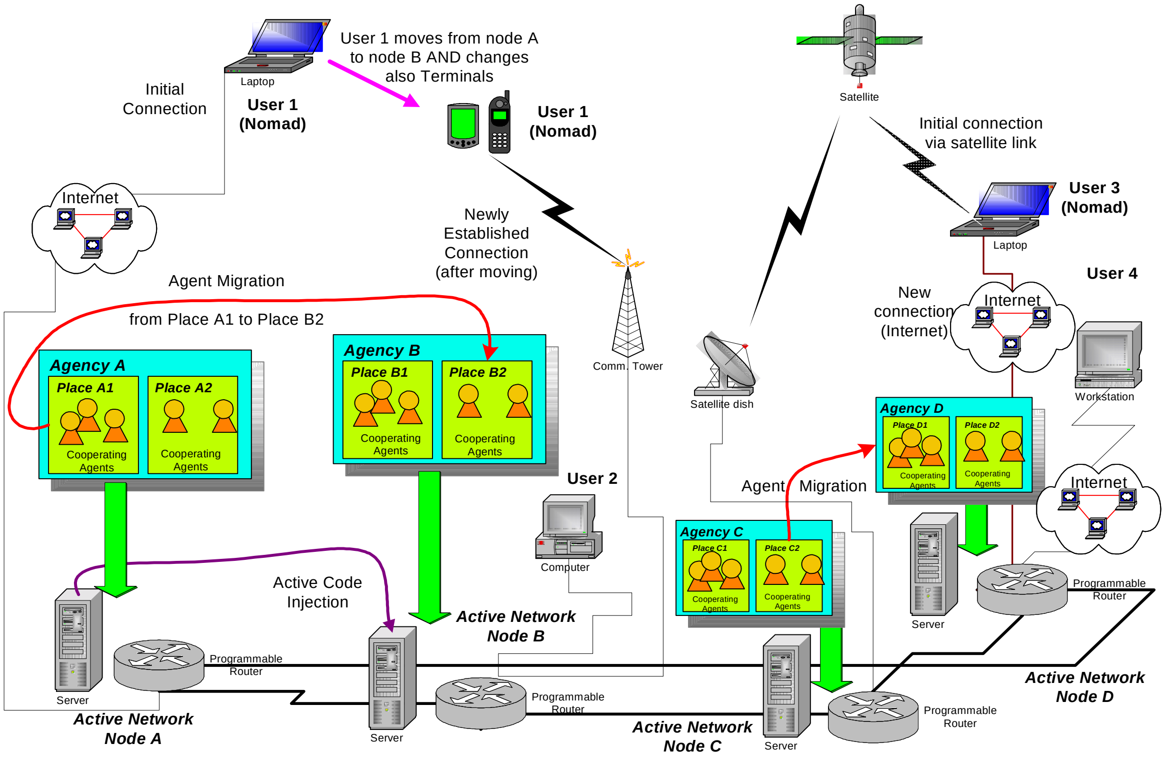

Fig. 4. Nomadic User Support Environment

VPN Selection of the Final Network Topology: The GA either returns and reports the possible solutions to the group of UAs or decides by itself for the best of the offered solutions. The final decision is made based on various facts e.g. connectivity bandwidth, error statistics, reputation of nodes contained in the topology, cost, security etc either by a human or the GA itself assuming that he is intelligent enough. Finally a network topology is selected and the final phase of agreement has to follow.

Deployment of the VPN and its services: The GA requests that service and network providers set-up the services and the network connections. When everything is done the GA informs the UAs that the VPN is ready for use. The GA either terminates at this stage or can be used as an central authority for future requests regarding the VPNs services and topology.

B. Dynamic VPN adaptation

After setting-up our VPN, group communication services can be deployed. An example infrastructure is that of Fig. 4 where users on domains A, B, C and D are connected. VPN members may span various network providers as some of them rely on the home network but others are roaming in foreign networks.

We will now examine what happens when the members are on the move. Lets suppose that a user that resided in domain A moves to node B and also changes his terminal from a laptop (advanced capability device) to a PDA (low capability device). The user had a teleconference in his laptop, which he wants to continue with the least possible disturbance in his new location (domain B).

The user's move typically in our infrastructure means that he has to be registered within the new domain and be provided at least with the same service quality as before. For that specific user, domain A is the home network and domain B is the serving network. The user movement implies that the agents providing the user with all the services move from execution place A1 in node A to execution place B2 in node B and resume their execution there after of course registering with the local nodes and adopt to the new environment. Although the steps are not strictly defined in such a scenario generally the following take place:

- The user is ready to move. This can be an automatic event (e.g. in a mobile device because the signal of the nearby communication tower is stronger) or a result of broken communication (e.g. the connection was terminated because of a communication hole or satellite technical problem). In any case the agents receive an "operations stop" event from the system agents.
- After receiving this "operations stop" signal the agents shutdown the services they provide to the user. They also notify the GA (which acts as a central information registry) that the current user will change its network position and all communication is temporarily suspended.
- The new destination address of the user in domain B is available to the local agents (still in domain server A). This can be done in advance (if the user move is normal)

or is sent to the agents the moment the user tries to log into the new visiting network B.

- Having obtained the new destination address, the agents migrate to the appropriate host in domain B where they are subject to the authentication control of the domain provider.
- Having successfully authenticated themselves (the agent code is signed by the user) they resume execution in the new node B in the visiting network.
- The user profile is consulted to see what are the services the user is subscribed to and how they should be personalized.
- Subsequently user's agents co-operate with the agents of the local node in order to retrieve the services supported by the node B for the visiting domain. The services that are the same with the home network are configured with the user's preferences and are activated. For the services that do not exist locally they ask local node agents of B to tunnel services from the home network. That assumes agent-to-agent communication and cooperation between the two domains A and B. If it is allowed by the policy of node B, the agents download the active code from a code server in the home environment that implements the missing services in domain B and install them on the node B. This is a very important step as it truly demonstrates the power and dynamicity of the VPN that is based on agent technology and active/programmable nodes. The possibility of downloading and installing code on the fly directly into the visiting environment is possible via the active networking technology in a flexible way.
- After having set-up everything they announce the new user place as the new part of the VPN network and inform the GA. Subsequently the GA can multicast the new VPN node to all affected UAs so that they configure the local services to comply with the new topology of the VPN.

At the end all this functionality is presented to the user (via the form of the services and automatic configuration). The effort is to have everything transparent and with minimal human intervention. For the end-user in our scenario it means that he can continue his teleconference uninterrupted as the agents on the back have taken care of this environment change. Code injection to a foreign network is not a trivial issue and is the driving force behind the active network community. However by mixing both the advantages the agents provide and the active networks promise, we can have a flexible solution for nomadic user support in VPNs.

V. CONCLUSIONS

Nomadic users are part of groups that have requirements that change unpredictably over time, especially due to the fact that they move constantly and spawn heterogeneous infrastructures. This kind of groups are interested in flexible VPNs that immediately adapt to environmental changes. Such groups can't use a basic version of a VPN to cover their requirements as they are not built in with mobility in mind and they are difficult and awkward in reconfiguration requests. Furthermore, those groups' lifetime is short and usually determined by some other external events. Such groups need to set-up and delete VPNs in minimal time. This flexibility to create and teardown such a virtual environment can be provided with the approach described above.

Agents can be used to deploy new services and program the nodes according to application's needs. By updating the underlying infrastructure's components on demand and by reusing in a Lego-like way the local services, agents are able to provide more sophisticated personalized services. Stationary agents that reside on the nodes not only offer their services but also respond to the environment changes - which may be unpredicted - by reconfiguring or updating node's components.

Both agent technology and active networks are research domains that continue to advance. Implementing the above functionality is not a trivial issue, especially due to the fact that the standardization process is not that advanced. One solution could be in adopting mobile agents and DPE based active networks. This solution is being investigated in one IST European project. There are some efforts (P1520 project [1]) to provide feedback to standardization bodies on programmable networks, but there are no standards so far available. As for agents, there are several standards from Object Management Group [4] and Foundation for Intelligent Physical Agents [3]. At the moment the only platform that complies with the MASIF [5] standard is Grasshopper [6].

REFERENCES

[1] IEEE P1520 Project Web Site : http://www.ieee-pin.org/

[2] Mobile Agents Links http://www.cetus-links.org/oo_mobile_agents.html

[3] FIPA Web Site: http://www.fipa.org/

[4] OMG Web Site : http://www.omg.org/

[5] MASIF - Mobile Agent System Interoperability Facility, http://www.omg.org/docs/orbos/98-03-09.pdf

[6] Grasshopper Mobile Agent System, http://www.grasshopper.de/

[7] S. Karnouskos, I. Busse, S. Covaci, "Place-Oriented Virtual Private Networks", HICSS-33, Jan. 4-7 2000, on the island of Maui, Hawaii.

[8] T. Campbell, H. G. De Meer, M. E. Kounavis, K. Miki, J. Vicente, and D. Villela, "A Survey of Programmable Networks", ACM SIGCOMM Computer Communications Review, Vol. 29, No 2, pg. 7-23, April 1999.

[9] Node OS Interface Specification, AN Node OS Working Group, Larry Peterson, ed., January 24, 2000.

[10] D. J. Wetherall, J. Guttag and D. L. Tennenhouse, "ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols", IEEE OPENARCH'98, San Francisco CA, Apr. 1998.

[11] D. Scott Alexander, "ALIEN: A Generalized Computing Model of Active Networks", Ph.D. Thesis, University of Pennsylvania, Dec. 1998.

[12] Mobile Agent Platforms http://www.informatik.uni-stuttgart.de/ipvr/vs/projekte/mole/mal/mal.html