Towards autonomic infrastructures via mobile agents and active networks

Stamatis Karnouskos

SAP Research, Vincenz-Priessnitz-Strasse 1, D-76131 Karlsruhe, Germany

ABSTRACT:

As we move towards service oriented complex infrastructures, what is needed is among others security, robustness and intelligence distributed within the network. Modern systems are too complicated to be centrally administered; therefore, the need for approaches that provide autonomic characteristics and are able to be self-sustained is evident. We present here one approach towards this goal, i.e. how we can build dynamic infrastructures based on mobile agents (MA) and active networks (AN). Both concepts share common ground at architectural level, which makes it interesting to use a mix of them to provide a more sophisticated framework for building dynamic systems. We argue that by using this combination more autonomous systems can be built that can effectively possess at least at some level self-* features such as self-management, self-healing etc which in conjunction with cooperation capabilities will lead to the deployment of dynamic infrastructures that autonomously identify and adapt to external/internal events. As an example, the implementation of an autonomous network-based security service is analyzed, which proves that denial of service attacks can be managed by the network itself intelligently and in an autonomic fashion.

INTRODUCTION

Systems and services are becoming more ubiquitous which yells for sophisticated solutions to be in place. As we move towards the "Internet of things" (Dolin, 2006), it can be expected that millions of devices of different size and capability will be connected and interact with each other over IP e.g. sensor networks (Marsh, 2004). Therefore, any approach will have to take into consideration that:

- Complexity will increase
- Heterogeneity in devices, software platforms, online services etc will increase
- A big proportion of end-nodes will be connected wirelessly to the backbone infrastructure (the line of wired vs. wireless systems will blur more)
- Bandwidth and computing power will increase
- Ad-hoc computing, collaboration, task delegation, and environmental adaptation will be a basic necessity
- On-demand software and service deployment will be vital

S. Karnouskos, "Towards autonomic infrastructures via mobile agents and active networks", in Encyclopedia of Internet Technologies and Applications, M. Freire, M. Pereira, Eds., IDEA Group Inc, 2007.

- Security and its satellite services will gain importance

In such an assumed future infrastructure, autonomic systems are expected to be of considerable help, since they will be able to be at a great degree self-sustained and also react to a dynamic changing environment.

Autonomic computing (Sterritt et. al, 2005) was introduced by IBM as a mean to target increasing computer system complexity and aimed initially at automating management of enterprise computational systems. In "The Vision of Autonomic Computing" (Kephart and Chess, 2003) it is stated that the dream of interconnectivity of computing systems and devices could become the "nightmare of pervasive computing" in which architects are unable to anticipate, design and maintain the complexity of interactions. The essence of autonomic computing is system self-management, freeing administrators of low-level task management whilst delivering an optimized system. In a self-managing system Autonomic System, the human operator does not control the system directly, but only defines general policies and rules that serve as an input for the self-management process. For this process, IBM has defined the following four functional areas:

- ***Self-Configuration:*** Automatic configuration of components
- ***Self-Healing*:** Automatic discovery, and correction of faults
- ***Self-Optimization*:** Automatic monitoring and control of resources to ensure the optimal functioning with respect to the defined requirements
- ***Self-Protection*:** Proactive identification and protection from arbitrary attacks.

There are two strategies in achieving autonomic behavior i.e. through adaptive learning and via integral engineering into systems (Sterritt, 2004). Our approach focuses on how to engineer such an autonomous system, while adaptive learning or self-learning is seen as an ad-hoc component that can be imported from the domain of intelligent agents.

AMALGAMATION OF ACTIVE NETWORKS AND MOBILE AGENTS

Active and programmable networks (Karnouskos and Denazis, 2004) introduce a new network paradigm where network-aware applications and services can be not only distributed, but also can configure the heterogeneous network to optimally respond to task-specific requirements. We are able to utilize within the network a) computation as we are able to compute on data received from active nodes and b) programmability, as we can inject user code into the network nodes in order to realize customized computation. Being able to achieve the above, we succeed in decoupling network services from the underlying hardware, deploy fine-grained customized services, relax the dependencies on network vendors and standardization bodies and generally open the way for higher level network-based application programming interfaces.

Agents 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 (Jennings and Wooldridge, 1996). One key characteristic of agents is mobility (mobile agents), which allows them to transport themselves from node to node and continue their execution there. Additionally, autonomy, independent decision making, goal directed behavior,

social ability are also key characteristics agents may possess (Genesereth and Ketchpel, 1994). 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 dependencies, fault tolerance etc. The number of mobile agent platforms coming from the commercial sector, as well as the academia is increasing day by day.

Active networks and Mobile Agent technology are very close to each other, sharing common ground in theoretical/conceptual as well as in implementation level. From the viewpoint of mobile agent research, existing active network approaches take mobile active code very close to the mobile agent paradigm.

- **Capsule**: a typical code mobility paradigm i.e. a single mobile agent

- **Active/Programmable node**: instantiation of code on-demand

From the active network research viewpoint, the mobile agent technology is one of the possible technologies that can be used to build active networks. Mobile agents are regarded as specific types of active code and a MA-based node as a specific type of active network node. Due to the fact that the MA research arena exists more than a decade now, it is far more advanced in active code related matters, therefore it could provide a boost to specific AN matters at conceptual and implementation level.

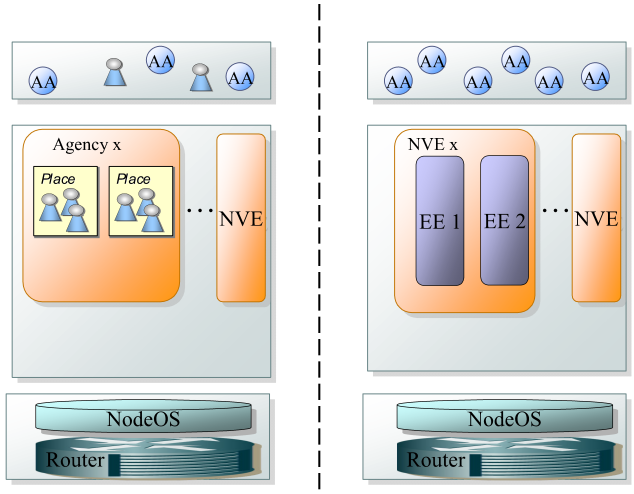

Figure 1 - The Agent-based AN node vs. the legacy one

The right side of Figure 1 depicts the architecture of a legacy active node, while on the right side the mobile agent based implementation is depicted. We can clearly distinguish the following levels:

- **Active Applications / Services** which exist as a result of execution of active code within an EE. An active code can a) provide a standalone service, or b) cooperate with other active codes residing on the same EE (EE-based service), on different EEs in the same node (multi-EE service) or even on different EEs in different nodes (network multi-EE service)

- **Execution Environments** where the active code executes. As an active node is expected to host multiple execution environments, these environments must have the ability to communicate with each other and to group in order to ease interactions. There are several functional types of EE aggregation such as Node Virtual Environment (NVE), Node Virtual Environment Network (NVEN), Execution Environment Network (EEN) etc.

- **NodeOS** which is an operating system for active nodes. The nodeOS provides generic services to the hosted EEs e.g. inter-EE communication (at EE, NVE or Active Application level), router resource management, EE isolation, etc. The nodeOS offers these services based on several facilities such as resource control, security, management, demultiplexing facilities.

As shown in Figure 1, one of the execution environments is the agent execution environment. This is the agency as described within the MASIF (OMG-MASIF, 1998) standard. The agent system consists 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. Cooperating agents reside in the agent-based EEs and via the facilities offered to them (re)-program the node. These can be either mobile agents (e.g. visiting agents) or even stationary intelligent ones that reside permanently in the EE implementing various services. The integrated approach of agents and active networks allows us to apply several security techniques at the network programming level (Karnouskos, 2001) that promote service and network security. Further info on this architecture and its security issues can be found in (Karnouskos, 2002). The mobile agent framework is able to realize the abstract functions of the EE, NVE etc. The AAs, are considered for implementation reasons to be mobile agents, but could also be applications that partially depend on them.

APPLICATION SCENARIO

An autonomic computing system (ACS) is able to (re)-configure itself in response to varying environmental conditions. Such a dynamic system can deal with unknown intrusions or attacks and is event able to recover from malfunctions or heal itself. The scenario presented here deals with a network based security system that is able to depict at some degree the characteristics of an ACS.

Securing a network nowadays is synonymous with hardening of its services. However, this approach makes the network inflexible and blurs the line between security and usability. Furthermore, each node has its own requirements on security which may also be varying in time. Within the vision of “Internet of things”, such per node or even per task modification of security, would be impossible to manage due to the large number and complexity of devices. Furthermore, those devices will build ad-hoc short-lived networks, where the burden of taking such actions may not be justified. Additionally, no common base exists among various security solutions available in the market. In other words, available products do not communicate with each other (interoperate) and work alone for their own and their distribution company’s good and not necessarily for the user’s network. A collaborative approach must be considered, however due to the nature of the future networks, this must be done on-demand and customized to the specific context. Community aware tactics on the other side may offer a better alternative. Adopting modeling approaches from the evolution of biological systems [Forrest et. al, 1997], they are seen as networks formed from cooperating living parts that interoperate at various levels and share information. ACS systems seem a promising approach towards that direction, mainly due to their self-* characteristics.

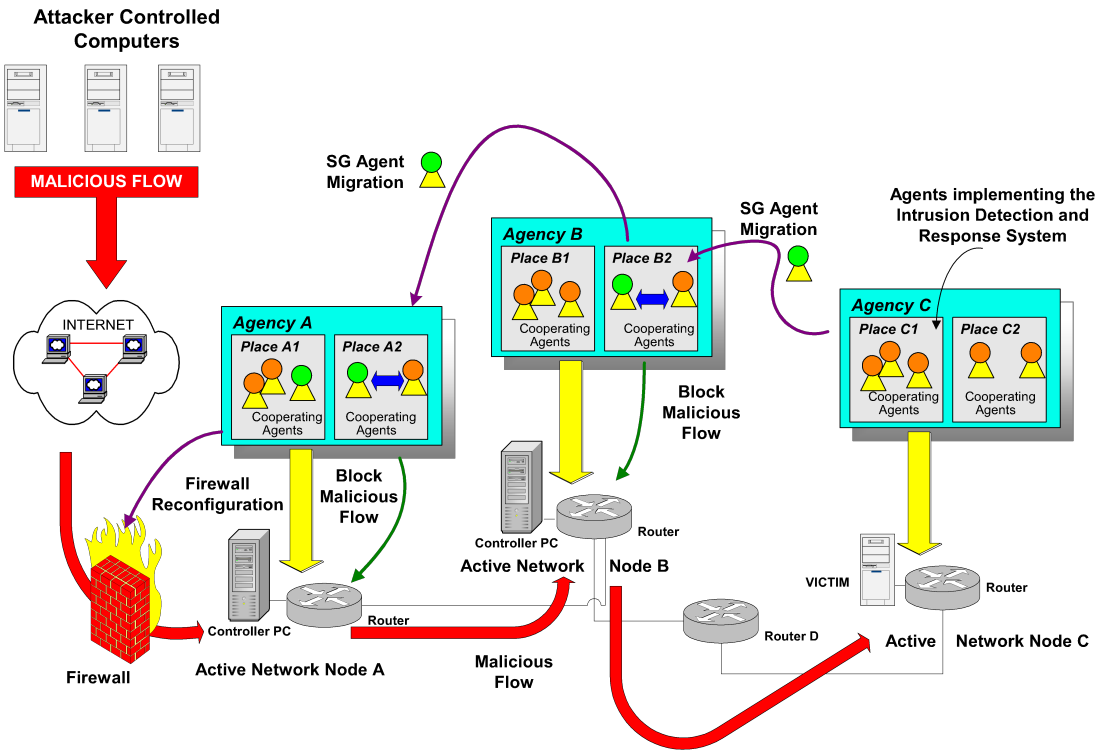

Figure 2 - DoS threat management

We assume a typical denial of service (DoS) attack scenario. As depicted in Figure 2, the network topology consists of various active nodes (e.g. nodes A, B, C) and legacy nodes (e.g. node D). In normal operation, the agents that implement our system reside within the agencies and filter the flow that is directed to the node. At some point the attacker initiates the DoS attack via the compromised hosts against the AN node C. One agent in node C detects the attack. This can be a result of an attack signature recognition (if the attack is known and exists in the system database) or a result of a dynamic correlation of events received by the system. Once the attack is detected, several security guards (SG) are released within the network (dynamic lookup of the neighboring nodes) and the attack info is disseminated towards the other nodes that reside within the path of the attack. In this way, the agents continue in an autonomic way to roam the network, identify the nodes that are prone to this attack and share info which eventually lead to a policy change and blocking of the specific malicious traffic within that node. At the end, the malicious flow is blocked towards the borders of the domain, and the network nodes are protected from this attack. Further detailed information about this approach can be found in (Karnouskos, 2004). The engines behind the data analysis and event correlation as well as decision and action management can be standalone; however, it is much more interesting if they act in a collaborative manner. Therefore, at each domain central analysis points (CAP) exist, which have the overview of what is happening in the domain, making therefore easier to recognize attacks that include multiple nodes in different parts of the network. CAPs have the global view and therefore are more efficient in attack recognition and decision making, while the action is done locally on each node; a tactic that allows thin components to be deployed even to devices that do not feature high computational capabilities.

The result of this approach, is that we have a network that features at some degree characteristics if autonomic systems. More specifically:

- ***Self-Configuration:*** Automatic configuration of the different components that recognize the attacks is done. The agents are goal-driven and are able to reconfigure themselves based on the environmental context they act on.

- ***Self-Healing*****:** Automatic discovery and correction of faults for network parts is done. Once this is detected the specific sub-network part can be isolated in order to avoid network misbehavior, and classical solutions to the problem can be applied.

- ***Self-Optimization*****:** Automatic monitoring and control of resources of the network can be done. In that case early indicators can be correlated and emerging problems are easier to be pinpointed.

- ***Self-Protection*****:** The network is protected from well-known attacks, including those that can be dynamically identified based on the correlation of events or even with "socializing" (i.e. information exchange) with other networks.

As presented, our approach deals with some aspects of ACS, while in the future more specific research should be invested towards a fine-grain exploitation of each of the features in detail. Self-managing mechanisms can have several instantiations e.g. self-governing, self-correction, self-organization, self-scheduling, self-planning, self-administration, self-optimization, self-monitoring, self-adjusting, self-tuning, self-configuration, self-diagnosis of faults, self-protection, self-healing, self-recovery, self-learning, self-sensing/perceiving, self-modeling, self-evolution, self-assessment of risks etc (Tianfeld and Unland, 2004).

CONCLUSIONS

We have presented an approach that is based on the amalgamation of active networks and mobile agents. We merge specific capabilities from each domain e.g. the network programmability from active networks and the autonomic, goal-driven social characteristics of mobile agents in order to create a powerful combination and implement a system that depicts at some degree behavior that characterizes autonomic systems. The approach taken is open and can be seen as a platform to further integrate research results coming from the two domains. Furthermore, we have not yet touched issues like self-learning mechanisms which however initially could be imported from the work done already by the research community on intelligent software agents. Security, trust and privacy issues as identified by (Cardoso and Freire, 2005) need to be further tackled, especially because our collaborative approach taken here heavily depends on them. Finally, other approaches that move towards the usage of agents for implementing specific scenarios also exist (Soldatos et. al, 2006).

The essence of autonomic computing systems is the creation of dynamic infrastructures that can deal in a proactive way with changing environment contexts; a fact that is gaining importance as we move towards a complex heterogeneous infrastructure e.g. as depicted in the "Internet of things" where all devices interconnected will form also ad-hoc networks for even task specific goals. Without such approaches, large-scale complex computing systems will be unmanageable.

REFERENCES

- Cardoso, R. and Freire, M. (2005). "Towards Autonomic Minimization of Security Vulnerabilities Exploitation in Hybrid Network Environments", in proceedings of the Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services (ICAS/ICNS 2005), Petre Dini and Pascal Lorenz (Eds.), October 23-28, 2005, Papeete, Tahiti, French Polynesia, IEEE Computer Society Press, Los Alamitos, CA, ISBN: 0-7695-2450-8.

- Dolin, R. A. (2006). Deploying the "Internet of Things". In Proceedings of the international Symposium on Applications on internet (January 23 - 27, 2006). SAINT. IEEE Computer Society, Washington, DC, 216-219.

- Genesereth, M. R. and Ketchpel, S. P. (1994). Software agents. Commun. ACM 37, 7 (Jul. 1994), 48-ff.

- Jennings, N. and Wooldridge, M. (1996). Software Agents. IEE Review 42(1), pages 17-21. January 1996. http://www.csc.liv.ac.uk/~mjw/pubs/iee-review96.pdf

- Karnouskos, S. (2001). "Security Implications of Implementing Active Network Infrastructures using Agent Technology", Special Issue on Active Networks and Services, Computer Networks Journal, Volume 36, Issue 1, pp 87-100, June 2001 (ISSN 1389-1286).

- Karnouskos, S. and Denazis, S. (2004). Programmable Networks: Background, in the book of A. Galis, S. Denazis, C. Brou, C. Klein (editors), "Programmable Networks for IP Service Deployment", Artech House Books, May 2004 (ISBN: 1-58053-745-6).

- Karnouskos, S., (2002). "Realization of a Secure Active and Programmable Network Infrastructure via Mobile Agent Technology", Special Issue on Computational Intelligence in Telecommunications Networks, Computer Communications Journal, Volume 25, Issue 16, pp. 1465-1476, October 2002 (ISSN: 0140-3664).

- Karnouskos, S., (2004) Community-aware Network Security and a DDoS response system, International Journal: Annals of Telecommunications (Annales des Télécommunications), Special issue on Active Networks, vol. 59 n°5-6, May-June 2004.

- Kephart, J. O. and Chess, D. M. (2003). The Vision of Autonomic Computing. Computer 36, 1 (Jan. 2003), 41-50.

- Marsh, D., Tynan, R., O'Kane, D. and O'Hare, G. (2004). Autonomic wireless sensor networks, Engineering Applications of Artificial Intelligence, Volume 17, Issue 7, Autonomic Computing Systems, October 2004, Pages 741-748.

- OMG-MASIF, (1998). Mobile Agent System Interoperability Facility, OMG, http://www.omg.org/docs/orbos/98-03-09.pdf

- S. Forrest, S. Hofmeyr, and A. Somayaji, (1997). "Computer Immunology" Communications of the ACM Vol. 40, No. 10, pp. 88-96.

- Soldatos, J., Pandis, I., Stamatis, K., Polymenakos, L. and Crowley, J. (2006). Agent based middleware infrastructure for autonomous context-aware ubiquitous computing services, Computer Communications, In Press, available online 17 Jan. 2006.

- Sterritt, R. (2004) Autonomic networks: engineering the self-healing property, Engineering Applications of Artificial Intelligence, Volume 17, Issue 7, Autonomic Computing Systems, October 2004, Pages 727-739.

- Sterritt, R., Parashar, M., Tianfield, H., Unland, R. (2005). A concise introduction to autonomic computing, Advanced Engineering Informatics, Volume 19, Issue 3, Autonomic Computing, July 2005, Pages 181-187.

- Tianfield,H. and Unland, R. (2004). Towards autonomic computing systems, Engineering Applications of Artificial Intelligence, Volume 17, Issue 7, Autonomic Computing Systems, October 2004, Pages 689-699.

KEY TERMS

- **Autonomic Computing**: It is an initiative started by IBM in 2001. Its ultimate aim is to create self-managing computer systems to overcome their rapidly growing complexity and to enable their further growth.

- **DoS**: Denial of service attacks have as a result that computers consume their resources for malicious events without being able to further processing legitimate user requests

- **Active networks:** Active networks are a communication paradigm that allows packets flowing through a communication network to dynamically modify the operation of the network.

- **Mobile agents:** A mobile agent is a composition of computer software and data which is able to migrate (move) from one computer to another autonomously and continue its execution on the destination computer.

- **Sensor networks:** Sensor networks are computer networks of many, spatially distributed devices using sensors to monitor conditions at different locations, such as temperature, sound, vibration, pressure, motion or pollutants. Usually these devices are small and inexpensive, so that they can be produced and deployed in large numbers, and so their resources in terms of energy, memory, computational speed and bandwidth are severely constrained.

- **Execution environment:** This is the place where the active code executes. The EE offers access to the core node resources via a policy-controlled scheme. This can be for instance a mobile agent system that takes care of the execution of an agent.

- **Active application:** This is the code that is actually executed in the Execution Environment of the node. Via its execution in the EE, the code programs the node according to user's preferences.