

# Towards a Policy-based Autonomic Virtual Network to support Differentiated Security Services

Steven Davy<sup>1</sup>, Claire Fahy<sup>1</sup>, Leigh Griffin<sup>1</sup>, Zohra Boudjemil<sup>1</sup>, Andreas Berl<sup>2</sup>,  
Andreas Fischer<sup>2</sup>, Hermann de Meer<sup>2</sup>, John Strassner<sup>3</sup>

<sup>1</sup>Waterford Institute of Technology, Cork Road, Waterford, Ireland  
{sdavy, cfahy, lgriffin, zboudjemil}@tssg.org

<sup>2</sup>University of Passau, Innstr. 43, 94032 Passau, Germany  
{andreas.berl, andreas.fischer, demeer}@uni-passau.de

<sup>3</sup>Motorola Labs, Schaumburg, IL, USA  
john.strassner@motorola.com

**Abstract.** This paper presents an approach to provisioning network services in an autonomic network, using virtualised routers as an enabler. The approach provides business users a method of describing the requirements and behaviour of a set of network services using policies, while abstracting the users from complicated network configuration tasks. It then dimensions a virtual network dedicated to provisioning these services. Virtualization of the network resources enables a modularised approach in which fault tolerance, redundancy and security concerns are catered for, specific to the service requirements. Moreover, it enables concurrent handling of the (possibly conflicting) needs of several different services. The paper describes an initial prototype implementation and a use case designed to illustrate the benefits of the approach.

## 1. Introduction

The management of communication networks is becoming increasingly difficult, particularly when faced with a changing Internet where more dynamic and flexible deployment of network services is required. Such services as, e.g., network security services, currently need to be carefully planned and configured to provide the highest level of reliability. This is often a slow and error prone task where configurations need to be carefully planned and rolled out. There are a number of promising techniques that enable management systems to abstract from the complexity and heterogeneity of the communication network. Autonomic network management aims to address the problems associated to current network management by pushing the responsibility of ensuring the proper operation of the network to algorithms and processes that exhibit “autonomic” characteristics (Jennings et al., 2007).

Towards reaching this goal, how to effectively manage a communication network according to business policies has to be investigated, so that available resources are used to their maximum potential. Virtualised network resources can abstract the heterogeneity of the network, and policy based network management can define the intended behaviour of the network via business policies, while abstracting from the complex configuration of individual devices. It is envisaged to employ resource virtualization techniques as mechanisms towards full service-aware networks. Virtualization is an abstraction from the underlying hardware and creates a virtual resource overlay. By engaging policy based management to manage this overlay, high level service behaviour can be described and the network will be automatically adjusted to exhibit the desired behaviour. In this way, physical resource usage can be maximised while

new capabilities can be dynamically introduced into the network to support new service requirements; thus reducing operation overhead.

This paper presents a first iteration of a policy-based virtual network management system, where policies are used to define when and where virtual routers need to be instantiated and configured to provision higher level network services, thus exhibiting autonomic characteristics in the process. Section 2 presents the Autonomic Virtual Network management system and a use case is presented concerning the deployment of network security services governed by business policies. Section 3 outlines the evaluation of the implementation to realise the use case. Section 4 describes the related work in the area of virtual networks and policy based management. Section 5 concludes the paper and describes future work on enhancing the implementation.

## **2. Autonomic Virtual Network**

A virtual network is an environment where each physical router in the network is running a Virtual Machine Monitor (VMM), featuring the ability to host one or several virtual machines on its hardware. These virtual machines take the role of virtual routers (VR). They use the underlying physical hardware to build up links with other virtual machines on other physical routers, forming virtual networks in the process. Being stored entirely in files or within a file structure, VRs can have different functionalities and capabilities, designed a-priori to deployment. Moreover, VRs are easily created, started, paused, stopped and even moved to other physical routers. In addition to common virtualization benefits, like consolidation of hardware, energy savings, or easy backup mechanisms which insure against hardware failures, the concept of VRs provides additional flexibility by allowing the autonomic restructuring of the logical network available to clients. This makes VRs a basic enabler to support different (possibly conflicting) requirements for different services at the same time. Two steps are envisioned. In the first step only border routers are virtualized, which already enables a set of new functions (e.g. the one described in the following use case). In the second step more demanding functions (e.g. quality of service functions) are enabled by virtualizing all routers of a communication network. By governing the virtual network with business policies, a degree of flexibility central to an autonomic network is achieved.

### **2.1 Implementation**

XEN (Barham et al., 2003) is used as the virtualization solution. The hypervisor XEN offers hardware to its guests, which is similar, but not identical to the underlying physical hardware. By using paravirtualization it achieves near-native speed for its virtual machines – a major feature considering the performance requirements of routers (and VRs) which have to handle large amounts of traffic in real time. XEN allows its guests to get direct access to certain parts of that hardware, possibly reserving some hardware resources for certain VRs. This allows, e.g., to assign each VR its own network interface card, without having to emulate the respective hardware. By using preconfigured VRs as blueprints, it becomes possible to introduce new features and functionality into the network without introducing unwanted side-effects to the legacy network, simply by distributing new VR images to the physical routers. Through an interface, environmental restrictions on the virtual network environment can be specified – e.g. VRs may only have access to a part of the available physical links, causing them to be logically isolated from each other although they may still be physically connected.

Also, other parameters like guaranteed bandwidth or processing power can be managed dynamically, allowing them to adapt to changing user needs.

In order to effectively manage VRs, a policy based management approach is taken, where business objectives are represented as policies and these policies are used to guide the behaviour of the virtual network. By elevating the definition of network behaviour to a high level of abstraction, advantage can be taken of the benefits of virtual networks to realise the behaviour specified by policies. This approach can only be realised by supporting the definition of policies with a rich system information model that describes the structure and relationships between managed entities in the network. Policies are deployed in a policy continuum, where business level policies define the behaviour of customer services, but do not specify the implementation of the services. These policies are translated into policies supplied to a rule engine (system level policies). Our rule engine is based on JBoss rules, which is an open source production rule engine. The process followed to build the policies for the system is described in Barrett et al. (2007) and outlined here. 1) The information model that represents the managed system is first tagged to indicate those concepts that are most aligned with business concerns. 2) The tagged classes and relationships are processed using model driven development techniques to produce domain specific languages (DSL). 3) The developed DSLs are used to describe the desired configuration of customer services along with their desired dynamic behaviour (in the form of policies). 4) The policies are translated into system rules that can be processed in the JBoss Rule engine. The rules are combined with system specific logic that can be used to interface with the managed virtual resources.

Once the system rules have been analysed and deployed, the PBM is run. The premise behind the runtime operation of a PBM is that events sensed from the changes in the network are used to trigger the evaluation of policies. For a policy rule to be fired, its conditions must be satisfied. Policy conditions are defined over the state of the network and may be associated to a range of attributes, for example, time of day, current set of users or network congestion levels. Once a policy rule has been triggered, evaluated and satisfied, it is then executed. The execution of a policy rule may take the form of a configuration update or in the case of virtual networks, the deployment of a new set of VRs. Essentially a control loop is created that enables the virtual network to self-configure, thereby exhibiting an essential autonomic property.

## **2.2 Use Case**

Companies are increasingly using the Internet to get communication services between different branch offices. To secure confidential communication between two branch offices, typically a Virtual Private Network (VPN) is set up. However, setting up and configuring a VPN correctly is a complicated task and prone to human error. According to Al-Shaer et al. (2004) IPsec configurations can conflict not only within a single router but across multiple routers. Moreover, encryption may not always be necessary, so having a VPN only for confidential data while sending all other data unencrypted may be a viable strategy for the company. One could imagine, for example, that transmission of highly sensitive data would occur over an encrypted channel, accepting some performance penalty due to the encryption, while less sensitive communication with strict QoS restrictions would use an unencrypted channel, trading secrecy for performance. To support such a scenario, mechanisms are required that allow to dynamically set up and/or tear down encrypted channels as needed. The approach shown in this paper uses the virtualization of networks to achieve the required dynamicity to be able to fully support this use case.

In our envisioned use case, the interaction between virtualization and legacy network infrastructure is handled by having the virtualized routers create IPv4 tunnels between them – e.g. an IPsec tunnel (see Figure 1). This provides some basic flexibility, as demonstrated in the use case below. However, an even greater amount of flexibility is reached, once the entire network consists of VRs. Such a situation would enable the extension of the scenario presented to more complicated scenarios, without the need to create specific tunnels for each service.

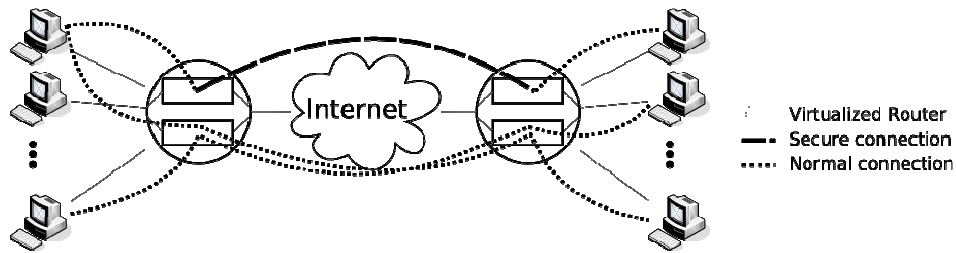


Figure 1: Secure and normal connections via virtual routers

A company has one main office and one branch office. Communication between the two offices is usually unencrypted. A new company policy specifies that data tagged as “classified” must not leave the company network unencrypted. Therefore, when such data is sent between the two offices, setup of an encrypted channel is necessary. In order to have the network set up the encrypted channel in accordance with business policies (i.e. autonomically), the following steps are carried out: 1) A new business policy is defined that requires “classified” traffic to be highly secure between the company networks. 2) The requirements for setting up secure tunnels are derived from policies defined by business users. Via the policy continuum, the business policies are translated into system policies that instruct the provision of configurations to support the realization of secure tunnels. 3) The border routers of both office networks get a new virtual image, providing routers that will tunnel all data through a preconfigured encrypted link. These VRs are initially paused. 4) Upon encountering such packets, the virtual border routers mentioned above are reconfigured to setup the secure tunnels and begin routing the packets through them. 5) Later the policy can be disabled and the virtual border routers are paused and replaced with default router images again to reduce performance penalties on the physical routers.

Table 1: Business Level / System Level Policies

Business Policy	System Policy
Secure “All Traffic” From “NetA” To “NetB” with Params: “Encryption=High”, “Authentication=Medium”	<pre> <b>If</b> (SecurityRequired &amp;&amp; Encryption=High &amp;&amp; Authentication=Medium) <b>Then</b> <b>DeployVR</b> [VR1, VR3]; <b>InstallIPsecKeys</b> on VR1,VR3 [   add {\$srcip}{\$dstip} esp 1 -E aes-ctr {\$keyesp} -A md5 {\$keyah}   add {\$dstip}{\$srcip} esp 2-E aes-ctr {\$keyesp} -A md5 {\$keyah} ]; <b>InstallIPsecPolicies</b> on VR1 [   spdadd {\$Edge1}{\$Edge2} any -P out ipsec esp/tunnel/{\$srcip}{\$dstip}/use]; <b>InstallIPsecPolicies</b> on VR3 [   spdadd {\$Edge2}{\$Edge1} any -P out ipsec esp/tunnel/{\$dstip}{\$srcip}/use];           </pre>

The requirements of the company are represented as business policies as outlined in Table 1. The business policies are described in a language amenable to the associated users, where key details need only be filled in; such as, whether all IP traffic is to be secured, or only traffic associated to a particular service. All key words and parameter options are modelled in the

information model of the system. The policies at the system level can be installed in to the specialised policy decision point rule engine. The specific system level policy described in Table 1 is triggered if there is a requirement for a secure tunnel connection with high encryption and medium authentication. In such case, the VR images with the appropriate capabilities are selected and IPsec policies are installed.

As the setup is performed by the proposed autonomic management functions the potential for errors due to manual configuration is reduced. Also, unlike the approach presented here, a conventional approach would require redundant and expensive hardware.

### 3. Evaluation

In this section the scenario described in the previous section is evaluated. Three Linux machines have been used as routers and XEN has been used as virtualization technology. A dedicated machine is located within the core of the network to function as a policy based management server; it has the capability to re-configure existing physical and VRs, and it can deploy new VRs on demand. To simplify the experimental setup, four clients are interconnected via three physical routers, as it can be seen in Figure 2.

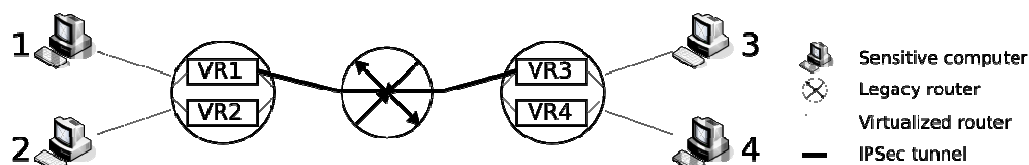


Figure 2: Implementation of the scenario

The legacy router in the middle represents the usual Internet routing. On the other two machines XEN has been enabled and two VRs each have been set up on top of it. VR2 and VR4 have been set up with normal routing functions, as originally provided by the Linux operating system. These two routers are used as default routers of the system. Once the policies have been described and deployed, they are triggered to instantiate and configure VR1 and VR3 to offer IPsec functionality. Clients 1-4 are ordinary Linux computers. All clients communicate with each other without using IPsec. Only the communication between client 1 and 4 is considered to be “classified” and has to be encrypted in the envisioned scenario. Therefore the routing tables of client 1 and client 4 are modified. If, e.g., client 1 wants to route to client 4 it does not use the default router (VR2) but it uses the IPsec enhanced router VR1. After setting up the scenario, the clients start to communicate with each other. As expected, communication between client 1 and 4 has been encrypted and been tunnelled by IPsec through the (simplified) Internet. All other communication took place without the use of IPsec via the default routers. The experiment has shown that the envisioned use case could be implemented by the use of VRs and the PBM. The original functionality of the network has been maintained, because the virtual default routers (VR2 and VR4) provided the same functionality than the real routers did before. The additional VRs (VR1 and VR3) brought new functionality into the network that has not been available before, without disruption of the original routing functions. It is reasonable to assume that further functions can be added to the network without disruption of its current functionality.

## 4. Related Work

### 4.1 Autonomic Network Management

Autonomic network management aims to alleviate the complexity associated to the management task of communication networks by introducing algorithms and processes at a lower level in the network that can operate in an “autonomic” fashion. The term autonomic comes from the autonomic nervous system of the human body and thus resulting inspired algorithms can exhibit self-\* features. Self-\* features (self-organising, self-healing, self-protecting, self-optimising) observed in these algorithms can enable complex management tasks to be governed at a higher-level. Policy-based management (PBM) systems are a promising enabler for autonomic communication as presented by Jennings et al. (2007). PBM is a concept developed originally for reducing the administrative complexity of reconfiguring a network due to changing business needs abstracted as policies. Policies are high level rules that describe what the network behaviour should be but not how it should be realised. How the behaviour is realised is down to the implementation of the PBM system; one such method is the use of the policy continuum as formalised in Davy et al. (2007). The policy continuum is a method of deploying high level business policies and relating them to lower level system policies that can be executed more easily.

Policy-based management systems have been extensively researched for use in domains from UMTS/Wireless network management (Zhang et al., 2003) to pervasive systems (Kagal et al., 2003). The approach to PBM in this paper is detailed by Barrett et al. (2007) in which a system information model is used to guide the development of policy languages that can capture the behaviour network services aligned to business objectives.

### 4.2 Virtualization

There are a number of different virtualization mechanisms mentioned in literature (Brendel, 2007). One way to classify virtualization techniques is to look at the layer where the virtual machine monitor (VMM) is located. The VMM is the virtualization software that multiplexes several virtual machines (VM) onto a single hardware instance. XEN and VMWare ESX-Server use the *full virtualization* scenario, where the VMM – also called *hypervisor* or *classical hypervisor* in this context – is located directly on top of the hardware and mediates access for the hosted VMs. While this approach promises increased speed as there is no additional layer between the VMM and the hardware, the VMM has to implement drivers for the underlying hardware in order to be able to handle requests from its hosted VMs. In the *hosted virtualization* scenario the VMM runs as a simple piece of client software within the host operating system (OS). In contrast to the previous approach, the VMM does not need its own drivers for all of the hardware, as it can use the facilities provided by the host OS. On the other hand, performance suffers, as system calls have to be handed through yet another layer. Also, the VMM will have to compete for available resources with other processes running on the host OS.

The x86-hardware platform poses some problems that would normally prohibit the use of a hypervisor based virtualization model (Popek et al., 1974). To solve this problem, the concept of *paravirtualization* has been introduced (Whitaker et al., 2002). It requires guest OSs to replace critical system calls by calls to the hypervisor, namely “hyper calls” – resulting, as a drawback, in a guest, which is no longer oblivious to the fact that it is running within a VM. The XEN hypervisor is a prominent representative of paravirtualization and has been chosen here to demonstrate the concept of VRs.

The term virtual router has also been in use for several other concepts. They have in common, that they do not refer to system virtualization (virtualization of a complete hardware) as it is used in this approach. Virtual routers are mentioned, e.g., in Campbell et al. (1999a) in the context of programmable networks. In the research of programmable networks (Campbell et al., 1999b) also reconfigurable networks, which support services, have been envisaged, but without basing on the concepts of virtualization, as it is done in this work. Also in Cole et al. (1998) the term virtual routers is used, for instance. In this approach multiple routers in concert create the illusion of a single virtual router.

## 5. Conclusions and Future Work

This paper presented an initial description and implementation of a policy-based virtual network, which is a potential enabler for autonomic network management. The approach combines the capabilities of policy-based management to abstract from the complex management tasks associated to network, with the benefits of virtual networks and in particular the use of virtual routers that can support increased flexibility with respect to the management of network resources. The use case presented concerned the dynamic re-configuration of the network to support a flexible VPN solution. The VPN is realised by rolling out a new virtual network topology with IPsec capabilities which was not originally supported in the network. The solution illustrates that new low level functionality can be rapidly introduced into the network in accordance to business policies with limited impact on the network.

Future work will be concerned with extending the capabilities of the implementation to deal with more complicated use cases, such as: stacked encryption challenges and secure quality of service.

**Acknowledgments.** Parts of the work in this paper were undertaken in the context of the AutoI project (Bassi et al., 2007, AutoI, 2007), EuroFGI (2006) and EuroNF (2007) – Networks of Excellence, which are partially financed by the EU. This paper was also partly funded by the German Research Foundation (Deutsche Forschungsgemeinschaft - DFG), contract number ME 1703/4-2.

## References

AutoI - Autonomic Internet Project, STREP, FP7, <http://www.ist-autoi.eu>, 2007

Al-Shaer, E. & Hamed, H. (2004), 'Discovery of Policy anomalies in Distributed Firewalls', in *Proc. of the 23rd Conf. IEEE Communications Soc. INFOCOM*, pp2605--2616.

Bassi, A., Denazis, S., Galis, A., Fahy, C., Serrano, M., Serrat, J., (2007) "Autonomic Internet: A Perspective for Future Internet Services Based on Autonomic Principles" - *IEEE 3rd International Week on Management of Networks and Services End-to-End Virtualization of Networks and Services (Manweek 2007) / MACE 2007 2nd IEEE Int. Workshop on Modelling Autonomic Communications Env.*, 29 October – 2 November, San José, California, USA

Barrett et al. (2007), 'A Model Based Approach for Policy Tool Generation and Policy Analysis', in *Proc. IEEE Global Information Infrastructure Symposium, GIIS*, pp99--105.

Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T. Ho, A., Neugebauer, R. (2003) Xen and the Art of Virtualization. *Proceedings of the nineteenth ACM symposium on Operating systems principles*, pp 164 – 177.

- Brendel, J.-C. (2007) *Linux Magazin – Technical Review – Alles über Virtualisierung, Ressourcenmanagement auf neuem Level*, Vol 1, January.
- Campbell, A.T., De Meer, H.G., Kounavis, M.E., Miki, K., Vicente, J.B., and Villela, D., (1999) A survey of programmable networks. *SIGCOMM Comput. Commun. Rev.*, vol 29, no. 2, pp7--23
- Campbell, A.T.; De Meer, H.G.; Kounavis, M.E.; Miki, K.; Vicente, J.; Villela, D.A. (1999) "The Genesis Kernel: a virtual network operating system for spawning network architectures," *Open Architectures and Network Programming Proceedings. OPENARCH '99.IEEE Second Conference on*, pp.115-127
- Davy, S., Jennings, B. and Strassner, J. (2007), 'The Policy Continuum - A Formal Model', in *Proc. of the 2nd IEEE International Workshop on Modelling Autonomic Communications Environments, MACE*, pp65-79.
- EuroFGI - Future Generation Internet, NoE, FP6, grant no. 028022, [http://eurongi.enst.fr/p\\_en\\_menu1\\_EuroFGIcom\\_368.html](http://eurongi.enst.fr/p_en_menu1_EuroFGIcom_368.html), 2006
- EuroNF - European Network of the Future, NoE, FP7, grant no. 216366, [http://euronf.enst.fr/en\\_accueil.html](http://euronf.enst.fr/en_accueil.html), 2007
- Jennings et al., (2007), 'Towards Autonomic Management of Communications Networks', *IEEE Communications Magazine* vol. 45, no. 10, pp112--121.
- Kagal, L., Finin, T. and Joshi, A. (2003), 'A Policy Language for a Pervasive Computing Environment', in *Proc. of the 4th IEEE International Workshop on Policies for Distributed Systems and Networks, POLICY*, pp63--74.
- Li, T., Cole, B., Morton, P., and Li, D., (1998) Cisco Hot Standby Router Protocol (HSRP), *IETF Network Working Group - RFC 2281*
- Popek, G. J., Goldberg, R. P. (1974) Formal Requirements for Virtualizable Third Generation Architectures. *Communications of the ACM*, Vol. 17, Issue 7, pp. 412-421
- VMWare ESX-Server, VMWare Inc, (2001) <http://www.vmware.com/products/vi/esx>
- Whitaker, A., Shaw, M., Gribble, S. D. (2002) "Denali: Lightweight Virtual Machines for Distributed and Networked Applications", *Technical Report 02-02-01, Univ. of Washington*
- Zhuang, W.; Gan, Y. S.; Loh, K. J. & Chua, K. C. (2003), 'Policy-Based QoS Management Architecture in an Integrated UMTS and WLAN Environment', *IEEE Communications Magazine* vol. 41, pp118--125.