

TITLE: EXTENDING MANTICORE TO MANAGE IP AND VIRTUAL MACHINE SLICES IN THE FEDERICA PROJECT

Alejandro Berna, Eduard Grasa, Sergi Figuerola

Fundació i2cat, C/ Gran Capità 2 - 4, Nexus I building, 2nd floor, office 203

08034 Barcelona, Catalonia, Spain.

e-mail: {alejandro.berna, eduard.grasa, sergi.figuerola} @i2cat.net

Abstract

FEDERICA will provide an IT infrastructure made of switches, routers and computing nodes that will allow the research networking community to test new protocols and techniques for the future Internet. MANTICORE is a software solution that delivers infrastructure as a service to router devices and IP networks. One of the main goals of FEDERICA is to allow network researchers to share the FEDERICA infrastructure without interfering each other's experiments. To achieve FEDERICA objectives the MANTICORE system will be enhanced with software routers and computer virtualization capabilities. This paper explains how the MANTICORE tool is being enhanced to satisfy FEDERICA requirements.

Keywords

Virtualization, Slice Management, Software Router, Virtual Machine

1. Introduction

Internet has been the world wide network that has changed the information flow and the communications around the whole planet. During Internet age the population has changed its lifestyle thanks to the new ways of communication it has created: e-mail, web sites, distributed applications, multimedia applications and so on. The number of users connected to the Internet is increasing exponentially. Currently the Internet population is near to 1.5 billion and it's becoming bigger quickly. Therefore the network has been upgraded with a lot of more uses and types of communications during its life (at the beginning only e-mail and web sites were used, currently it is possible to watch a streaming movie, do a videoconference, play games online...). On the other hand, the protocols and the different techniques to create this communication have not been upgraded as the use of Internet has. For example, the IP protocol that was created at the beginning of Internet is still used. The need to upgrade these protocols and techniques for the current requirements that the Internet users have is becoming a critical issue. The problem comes when a new protocol or technique is going to be tested: it is not possible to test something without having a risk of failure. If a test over the Internet network fails, it is possible to cause disconnections over the world. This risk is too big to be taken. There is a solution that can avoid this risk: having another infrastructure similar to the one used for Internet to perform the tests avoiding risks in the Internet infrastructure. However, with this solution, another problem appears: creating a new infrastructure similar to the Internet is really expensive. If every organization who wants to test new protocols needs to build its own infrastructure to perform the tests, every organization needs to invest a large amount of money. This big investment can be avoided by building and deploying a shared infrastructure permitting the option to share also the costs. Although this solution minimizes the investment in infrastructure, it introduces new problems and challenges due to the fact that the shared access to the infrastructure must be coordinated: the infrastructure administrators must ensure that a given set of resources are only used by a single organization at the same time. A solution to maximize the utilization of the resources and to allow parallel tests in the same infrastructure is creating independent slices. This means splitting the infrastructure into slices, portions, which work as if they were independent. For example, split a physical router in three slices and every slice has to work as if it was a physical independent router (this can be considered as an independent slice).

The main objective of the FEDERICA project (Federated E-infrastructure Dedicated to European Researchers Innovating in Computing network Architectures [1]) is creating an infrastructure similar to the one used in Internet allowing the end users to perform independent tests of new protocols and applications. Thus the resources of this e-Infrastructure can be allocated to multiple independent slices and assigned to different network researchers. FEDERICA needs to allow the researchers to have the complete control of the resources in their slice, enabling disruptive experiments. During the first stage of the FEDERICA project, slices will be manually created and managed. While this procedure is acceptable

at the beginning, it imposes a series of limitations and disadvantages both to the FEDERICA NOC, responsible for creating the slices and managing its permissions, and to the FEDERICA users (the researchers that will configure the slices to run their Future Internet experiments): the FEDERICA NOC has to maintain a database of users and passwords, they have to use different command line interfaces and/or vendor tools to configure the network and computer resources, they have to manually maintain a list of what resources have been assigned to what users and so on. At the same time, FEDERICA researchers would have to maintain a list of the resources on their slice, and would also need to manually configure them. Therefore, a tool to create and manage infrastructure slices would simplify the operation and use of the FEDERICA e-Infrastructure.

Currently there are some tools that are being used to virtualize and manage network resources from an infrastructure. In this paper the tool that is going to be studied is the MANTICORE tool. MANTICORE is a tool that provides a Web Services based Resource Management System to control logical IP networks. This system can control (logical) routers devices (the current release based on Juniper [2] architecture). In order to fit in the FEDERICA scope, MANTICORE needs to be enhanced to manage and control all the devices that will compose the whole infrastructure. Specifically, MANTICORE needs to be enhanced to manage servers with some computer virtualization software and software based routers. These new capabilities are going to be explained in this paper.

This paper is structured as follows. Section 2 gives a general overview of FEDERICA and its main concepts. Section 3 explains the current version of MANTICORE tool. Section 4 describes the different enhancements in order to include MANTICORE within the FEDERICA scope. Finally section 5 describes the conclusions.

2. Scenario: The FEDERICA e-Infrastructure

FEDERICA is a physically distributed infrastructure and will allow end users to have the control of independent slices of this infrastructure in order to perform tests for the future of the Internet.

The FEDERICA infrastructure is being deployed through the whole Europe. It is composed by routers, switches and servers. The goal is creating an infrastructure that works as close as possible to the actual Internet infrastructure. As a first stage, a core is being deployed and, around this core, some edge nodes will be connected. Using a backbone network (the core) and distributed nodes around it (edge nodes) it is possible to assure that this network will behave similar to the Internet. Figure 1 depicts a simple example of an Internet infrastructure which is similar to the one used in FEDERICA.

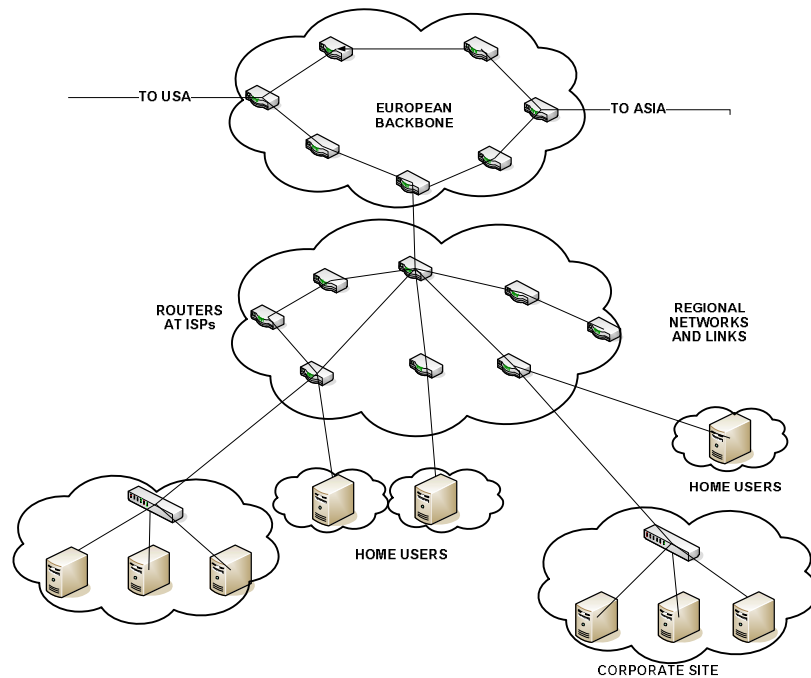


Figure 1 Example of Internet infrastructure

In the Figure it is possible to see the European Backbone (equivalent to the core of FEDERICA) and the connections to USA, Asia and to the regional network and links (the edge nodes in FEDERICA). Figure 2 provides an overview of the actual FEDERICA infrastructure.

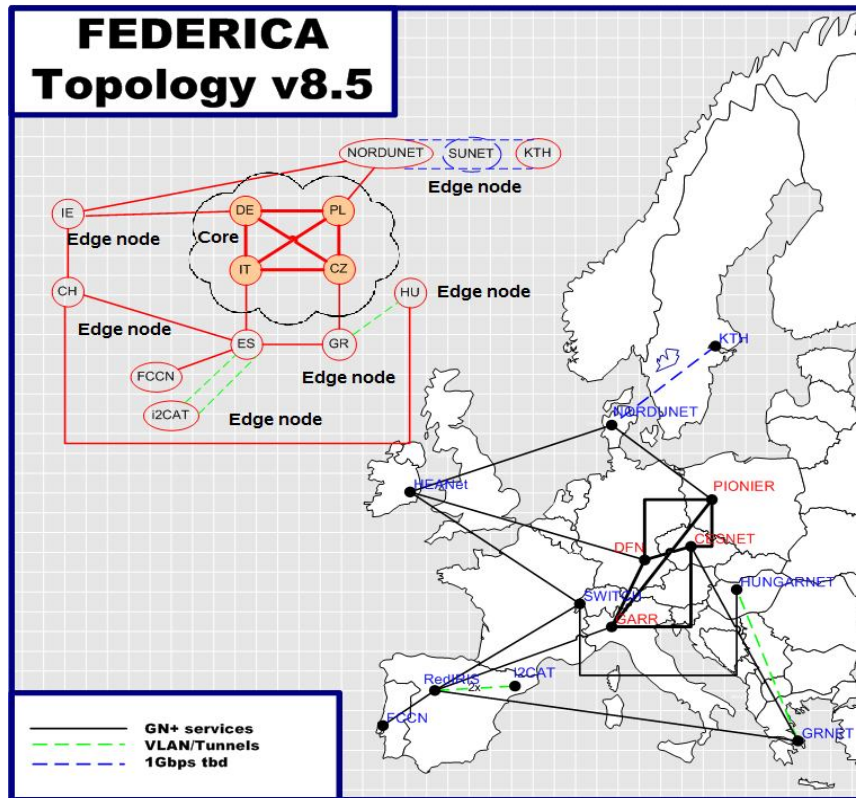


Figure 2 FEDERICA topology

All the devices that will compose the FEDERICA infrastructure will be sliced using different techniques. One of the greatest challenges of the project is to specify which technique will be used to create slices of each kind of devices (routers, switches and computers). Once these techniques have been specified, they will be used to develop a list of services (one to create slices from routers, another from computers and so on). Summarizing, FEDERICA will offer the possibility to end users to rent for a limited time slices of an infrastructure similar to the one used by the Internet. Having the control of this slice the end user will be able to test new protocols, techniques and applications that may solve the current Internet problems.

Routers will be sliced with the concept of logical routers; a logical router is a partition of a physical router. When a logical router is created, all the hardware's functionality is replicated creating different routing domains within a single physical router. For example, the routing table and the routing protocol daemons are replicated for every instance of a logical router created in the physical router. Thanks to this, it is possible to configure different protocols in different logical routers sharing the same physical device and not interfering between them. Specifically the technique which is going to be used is the one implemented by Juniper networks.

Switches have been widely virtualized during the past years; the technique used to create this virtualization has been Virtual LAN (VLAN). Virtual LAN (VLAN) creates level 2 virtual circuits over the Ethernet infrastructure. While with VLAN technology it is possible to isolate some ports of the switch from each other, it is not the same as fully virtualizing the Ethernet switch: a VLAN only isolates some interfaces from the others; it does not create independent management domains and control agents at the switch. In FEDERICA, the protocol IEEE 802.1Q will be used to manage VLANs.

The last technique, used to virtualize computers, is the technique used during all the past years: the hypervisor. A hypervisor is a virtualization platform that allows running different operating systems in the same host at the same time. Every instance that is running an operating system is called a Virtual Machine (VM). A hypervisor also virtualizes the hardware of the host; for example, if the host has one

network interface card (NIC), it is possible to generate different virtual NICs bridged to this physical one. Then it is possible to assign these different virtual NICs to different VMs. Also the CPU, the RAM memory, the CD-ROM, the hard disk or the USB port can be virtualized and shared by different VMs. There are two main types of hypervisor: native hypervisors or hosted hypervisors. The former is the one that directly runs on a given hardware platform as an operating system control program. The latter runs within an operating system environment. The one that will be used in FEDERICA is the native hypervisor. The reason is very straightforward: native hypervisors have better performance than hosted hypervisors. The hypervisor selected to manage the virtual machines is VMware ESXi [3]; it has been selected because it has good performance, it has fewer hardware limitations than other tools and it has a good remote management API.

FEDERICA will also manage software routers; a virtual machine with some software installed that makes it work as a router. A software router in FEDERICA will be a virtual machine with Ubuntu Server [4] as the main OS and Xorp [5] as the routing tool. Xorp has been selected because it is open source, supports a lot of protocols compared to other similar tools and has an easy way to add new protocols to test. All these kind of tools are managed by a command line interface (CLI).

3. The MANTICORE tool

MANTICORE is a software solution that delivers infrastructure as a service to router devices. It allows infrastructure owners to manage their physical as well as logical routers and to enable third parties to control them. MANTICORE also provides tools to assist infrastructure users in the creation and management of IP networks using the router resources of one or more infrastructure owners. The FEDERICA use case can be seen as an example of infrastructure as a service: the infrastructure owner (the FEDERICA NOC) enables infrastructure users (the network researchers) to get a subset of the available infrastructure (also known as slice) and configure it the way it better fits their needs. Therefore MANTICORE is a tool that can be potentially used to create and manage FEDERICA slices. However, the current MANTICORE implementation only supports Juniper based physical and logical routers, and the FEDERICA infrastructure is composed by Juniper routers, software based routers and computers.

3.1. Architecture

MANTICORE software architecture is based on the IaaS Framework [6] software. The IaaS framework is a set of resources, libraries and tools licensed under the Apache Software License version 2 that enable developers to quickly create new virtualization solutions based on the Framework programming model. The functionalities provided by these tools allow a developer to choose which web service stack will be used to expose the physical infrastructure as a service (supported SOAP engines include Axis2, CXF and Spring-WS), and provide a series of modules to plug-in capabilities like security, reservation management and data persistence to the infrastructure service. The Framework also provides libraries to speed up the development of drivers to communicate with the physical devices, like protocol parsers (TL1, NetConf), transport handlers (TCP, SSL, SSH) and a driver architecture called the IaaS Engine.

MANTICORE follows IaaS Framework architecture because thanks to its strict architecture, Web Services (WS) communication interface and modular engine core (which communicates with the different devices) it is possible to make a modular tool which is portable to other vendor devices. MANTICORE particular software modules are the Router WS and the IP Network WS. The former offers services to manage the device and the latter provides an API to create and configure an IP Network.

The Router WS is used to perform virtualization operations (for example creating a logical router) or to perform IP configurations (for example configuring OSPF) on logical or physical routers. It translates the abstract operations of its interface to messages that each particular routing device can understand. This job is achieved by means of the IaaS Engine. This engine acts like a driver that translates the operations and information of the abstract router model used by the Router WS to the protocol and parameters that the physical device can understand. The first MANTICORE implementation provides a driver for Juniper routers only, which are configured through the NetConf API [7]. NetConf is a device independent XML protocol designed to allow software agents to send configuration messages to physical devices (such as switches or routers) in a more structured way than through the use of Command Line Interface (CLI) languages, which are designed for the interaction between a human user and a physical device.

On the other hand, every IP Network WS instance represents a set of Ethernet Resource Services, which represent the ports of the physical or logical routers that this IP Network can control. The IP Network interface is used to perform IP configurations; it receives service requests and translates them on service requests to the Router WS. It provides the following capabilities to its clients:

- General configuration of the router interfaces. The user can configure the IP address, the network mask, the status (up or down) and other general parameters of the interface (represented by the Ethernet Resource Web Services).
- Add/Delete static route. The user applies a static entry to the routing table. This static entry can be for one or more source interfaces.
- Configure an IGP on an interface or the whole network. The user of an IP Network can apply a dynamic routing protocol (initially OSPF and RIP) to every interface inside it, and modify the IGP parameters.
- Configure an EGP on an interface or the whole network. The user can decide which other IP Networks he wants to peer with, and choose what routes are advertised to external networks. Simple BGP configurations will be generated (iBGP will also be configured between BGP nodes of the same AS).

The next Figure shows how these different modules are related including the WS and the communication protocols. It also shows a client for the WS, a graphical user interface (GUI client).

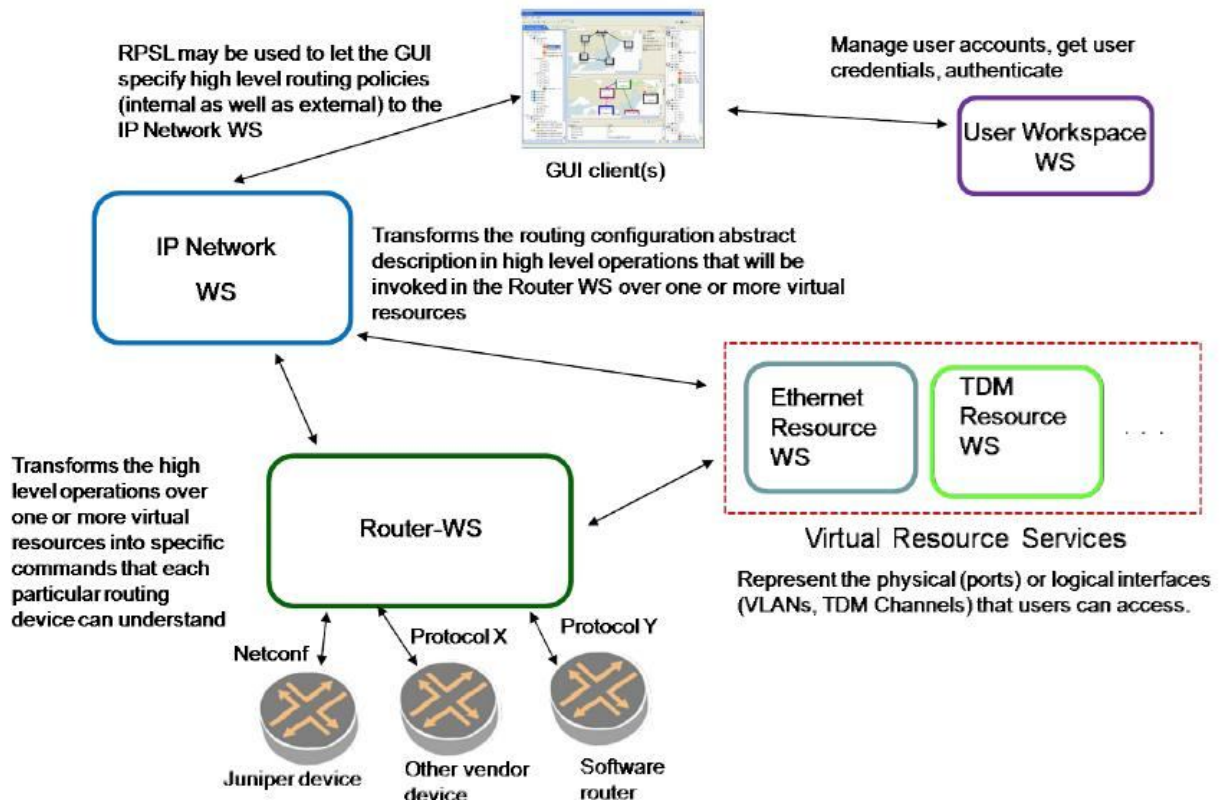


Figure 3. MANTICORE Software Architecture

A more detailed explanation is given in [8].

4. MANTICORE extensions

In order to use MANTICORE within the scope of FEDERICA it is necessary to enhance it to support the whole infrastructure composed by Juniper routers, software based routers and Sun servers. Currently MANTICORE is capable to manage and configure Juniper routers. Therefore MANTICORE will be enhanced with software based routers and Sun servers management capabilities.

4.1. MANTICORE virtual machines extensions

The first new capability that MANTICORE needs to fit in FEDERICA is the computer virtualization

capability. This new capability will allow MANTICORE to manage and control the Sun servers with the VMware ESXi tool (the virtualization platform chosen for the 1st FEDERICA release, see chapter 2) installed in the FEDERICA infrastructure. Specifically it will allow managing virtual resources of the server and, also, creating and managing virtual machines.

To manage VMware ESXi remotely there are a few potential ways; the best one is using VMware VI API because it is a Web Service based interface and it provides an object orientated architecture. However, some features need to be implemented by using some Linux commands (it is possible thanks to the Linux based OS that manages VMware) because the VMware VI API is not powerful enough. The MANTICORE architecture forces to define a transport and protocol for the different management APIs that need to be used: to run Linux based OS operations the transport and the protocol are SSH and Linux CLI. For the VMware VI API operations the transport and the protocol are HTTP and SOAP.

Moreover a data structure needs to be defined in order to control and manage the different servers of the FEDERICA infrastructure with the MANTICORE tool. This data structure needs to represent the different available resources and, also, the virtual resources generated in the server; the virtual hard disks, the virtual machines and its features, the resource pools (virtualized RAM and CPU) and the virtual NICs. The next Figure shows this data structure:

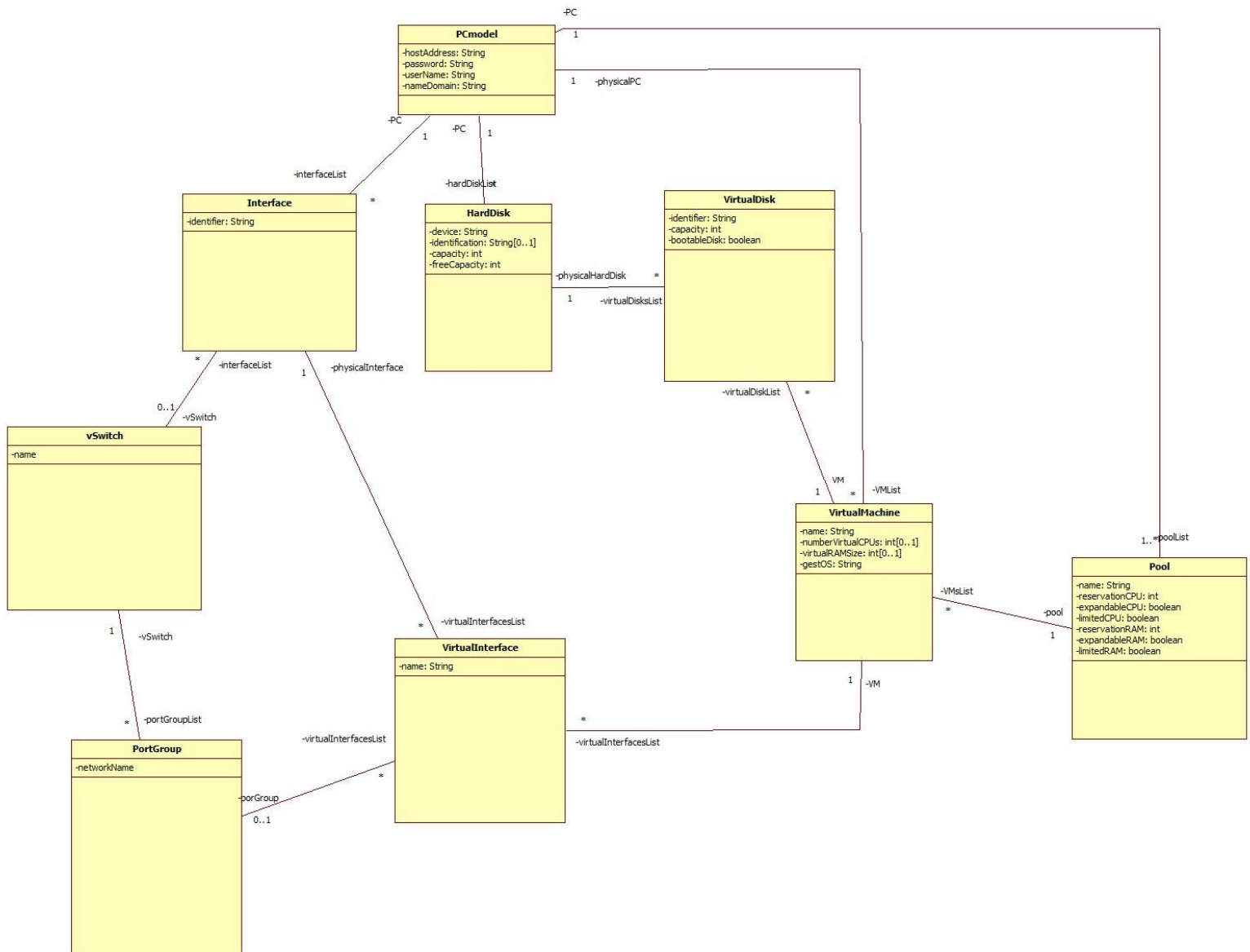


Figure 4. Computer data structure

With these different modules MANTICORE is capable to control and manage the FEDERICA servers. However, managing and controlling means to perform specific actions. These specific actions (necessary for FEDERICA requirements) are to copy templates of VMs, create blank VMs, change the hardware configuration of VMs, configure the remote access to these VMs so that the user can install their software, configure different network port groups (the way that VMware ESXi virtualizes the network interfaces), configure/create different resource pools and create/configure data stores (the way that VMware ESXi specifies local or remote hard disks available for virtualization purposes). The two main capabilities that will allow the NOC admin to create independent slices of FEDERICA infrastructure based on the end user requirements are:

- Copy templates of VMs: a VM template is a VM with software preinstalled and with the exact resources to run this software. For example, a software router template is a VM with Ubuntu Server installed as the OS and the Xorp tool installed as the routing tool. This template does not have any virtual interface (it's not necessary to run Ubuntu and Xorp) and has the exact virtual hard disk size to install Ubuntu and Xorp. When the end user requests a software router the NOC admin copies this template specifying the new hardware configuration (the virtual NICs, the virtual hard disk size, the virtual RAM size...) that the end user has defined in the request.
- Create blank VMs configuring remote access and allowing the end user to install the software he wants: the user requests a new VM with a specific virtual hardware configuration. The user also has previously created an iso image with the OS and the software that he wants to install in the blank VM. The NOC admin creates this new blank VM with the virtual hardware specification defined by the end user. He also configures remote VNC access to the virtual machine in order to permit remote management by the end user. Finally the NOC admin associates the iso image created previously by the end user as the main CD-ROM of the VM. The end user accesses his VM using VNC and installs all the software stored in the iso image. What the user gets at the end of the process is a VM with the hardware that he requested and also with all the software that he wanted to install.

Finally two more steps are necessary to finish enhancing MANTICORE with the computer capability: creating the Computer WS, capable of transforming services requests into messages that VMware ESXi can understand (in the first release, in next releases it can be another virtualization platform), and enhance the MANTICORE GUI with the new computer device view and also with all the wizards to perform the needed operations.

4.2. MANTICORE software routers extension

As explained on past subchapter, a software router will be created copying a software router template, a virtual machine template with Ubuntu Server and Xorp installed, with the MANTICORE computer capabilities.

The tool to be managed is the Xorp tool, therefore the communication needs to be done against it. Xorp offers a Command Line Interface (CLI) to manage it which needs to be used locally. Thus it is necessary to connect remotely with the Ubuntu Server and, once Ubuntu Server has been connected, call the Xorp CLI management interface. In order to communicate with the Ubuntu Server (from where Xorp CLI is used) it is enough creating a SSH connection. Merging these two concepts it is clear that the transport is SSH (a different one than the one used for Juniper routers) and the protocol is the Xorp CLI management interface.

The different actions that are going to be implemented are the same as the implemented for logical Juniper routers in MANTICORE: static routes, OSPF and BGP.

The data structure used to represent software routers needs to be the same that represent currently routers in MANTICORE (it needs to be portable to any vendor device). Hence this structure needs to be reviewed, and enhanced if it is necessary, in order to represent software based routers. Following this idea, the Router WS and the GUI need to be reviewed (and changed if necessary) to represent and permit the control of software router based.

4.3. MANTICORE Architecture Enhancements

As explained in the previous subchapters, MANTICORE will be enhanced with computer virtualization and software routers capabilities. This means that 2 news engines (software to communicate with specific vendor devices) will be created: one for VMware ESXi and another one for Xorp installed on an Ubuntu Server. Moreover the Computer WS will be added to MANTICORE (also the VM Resource WS equivalent to the Ethernet Resource WS used by the Router WS). Updating Figure 3 with the new enhancements creates the architecture depicted on Figure 3.

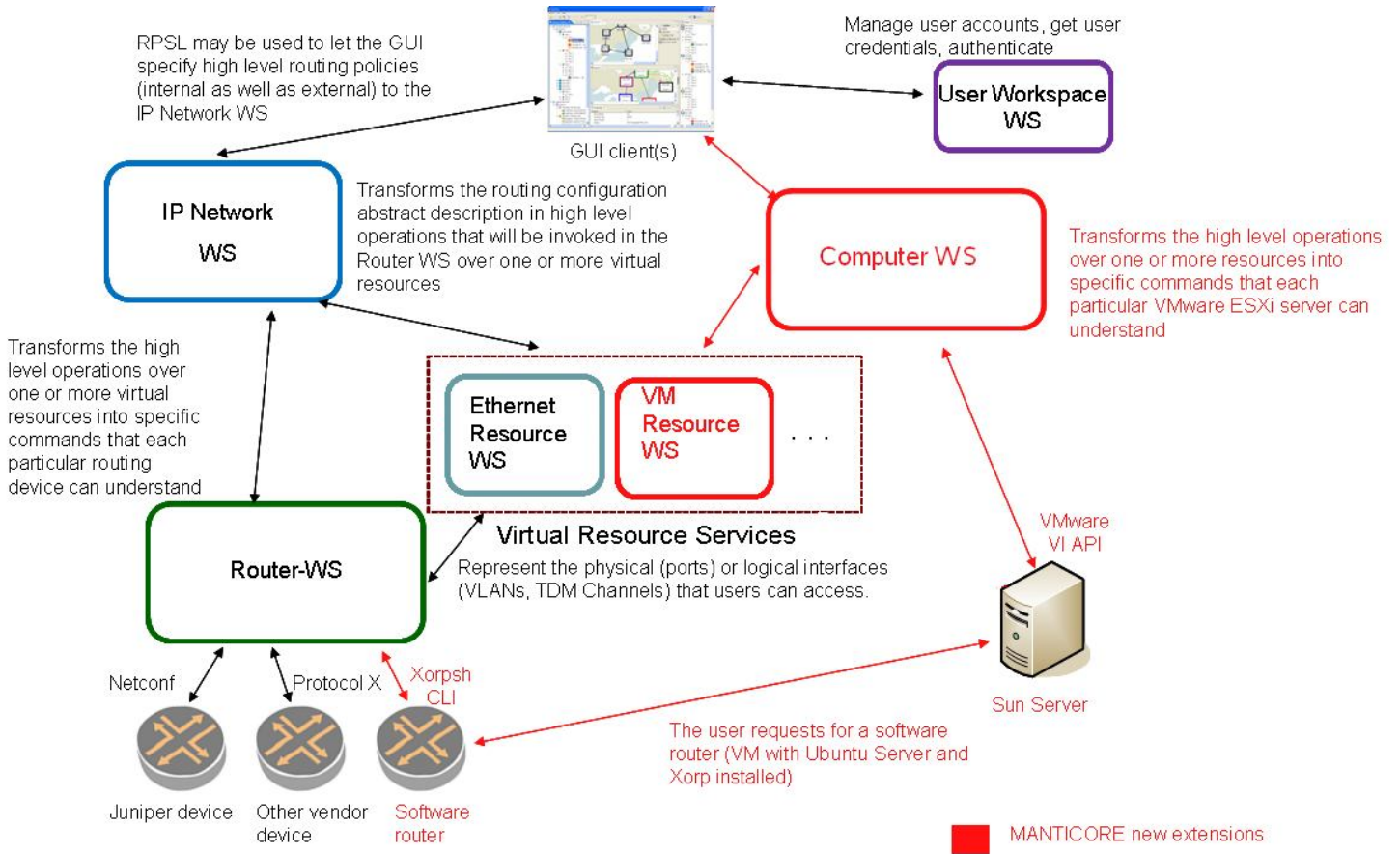


Figure 5. MANTICORE Software Architecture Extensions

A first implementation of MANTICORE extensions is currently being developed under the scope of FEDERICA SA 2.4 and JRA 2.3 activities.

5. Conclusions

The first conclusion obtained from the work explained in this paper is that it is possible to manage, control and virtualize the different devices that compose FEDERICA (Switches, Routers and PCs). It is possible to deploy an infrastructure similar to the one used by Internet and share this infrastructure between different researchers. These researchers can perform their networking experiments without the risk of interfering each other.

Moreover it has been shown that it is convenient to have a tool that manages all the resources from the FEDERICA infrastructure; it is necessary to maintain a database of all the devices and its configurations as well as lists with all the virtual slices and the end users to whom are subleased. Also it has been concluded that the current MANTICORE tool with some enhancements (computer virtualization and software based routers capabilities) can manage all these devices. Particularly MANTICORE will be capable of managing all the physical devices (create virtual slices and sublease them to end user, NOC capabilities) and IP virtual networks composed by logical routers and software based routers (configure IP settings of the IP slice, end user capabilities).

Acknowledgements

The FP7 project FEDERICA is partially supported by the European Commission under the Grant Agreement No.: RI-213107. The authors acknowledge the contribution of all project partners and specifically for his input Josep Pons (i2CAT).

References

- [1] M. Campanella: "Federated E-infrastructure Dedicated to European Researchers Innovating in Computing network Architectures", Future Internet Conference, 2 April 2008, <http://www.fp7-federica.eu/>
- [2] <http://www.juniper.net/>
- [3] <http://www.vmware.com/products/esxi/>
- [4] <http://www.ubuntu.com/>
- [5] M.Handley, O. Hodson, E. Kohler: "XORP goals and architecture", ACM SIGCOMM Hot topics in Networking 2002, <http://www.xorp.org/papers/xorphotnets.pdf>
- [6] M. Lemay: "An Introduction to IaaS Framework", August 2008, <http://www.iaasframework.com/>
- [7] NETCONF Configuration Protocol, RFC 4741, December 2006. Available online at: <http://www.ietf.org/rfc/rfc4741.txt>
- [8] E. Grasa, V.Reijs: "The MANTICORE Project: Providing Users with a Logical IP Network Service", TERENA Networking Conference, May 2008

Biographies

Alejandro Berna is a Graduate in Informatics Engineering of the Technical University of Catalonia (UPC, January 2008). In 2006 he joined the Telematics Engineering Department of the UPC where he collaborated in ENIGMA2 project until May 2007 and from July 2007 to December 2007 in MANTICORE project. In April 2008 he joined i2CAT Foundation to work on FEDERICA project.

Eduard Grasa: is a Graduate in Telecommunication Engineering of the Technical University of Catalonia (UPC, July 2004) and Ph.D. (UPC, February 2009). In 2003 he joined the Optical Communications Group (GCO), where he did his thesis on software architectures for the management and control of sliceable networks. During his thesis elaboration he collaborated with the Network Technologies Cluster of the i2CAT Foundation, which he joined in September 2008. He has participated in several national and international research projects, including UCLPv1, UCLPv2, HULP, FP6 PHOSPHORUS, FP7 FEDERICA, DREAMS, HPDMnet, IaaS Framework, Argia and MANTICORE.

Sergi Figuerola is the Coordinator of the Network Technologies Cluster of the i2CAT Foundation since 2004. A graduate in Telecommunication Engineering of the Technical University of Catalonia (UPC, January 2002), he holds a Masters in Project Management from La Salle (Universitat Ram3n Llull- 2004). He joined the UPC's Optical Communications Group (GCO) in November 2000. Since 2002 he is a PhD candidate of the Signal Theory and Communications Department, awarded with a PhDs fellowship by the Spanish Science and Education Ministry in 2003.