The MANTICORE project: Providing users with a Logical IP Network Service

Eduard Grasa , Xavier Hesselbach , Sergi Figuerola
Fundació i2cat, C/ Gran Capità 2-4, Edifici Nexus I, 2º planta, desp.203, 08034 Barcelona, Spain
e-mail: sergi.figuerola@i2cat.net, eduard.grasa@i2cat.net, xavierh@entel.upc.edu

Victor Reijs, Dave Wilson
HEAnet, 1st Floor, 15 George's Dock, IFSC, Dublin 1, Ireland
e-mail: victor.reijs@heanet.ie, dave.wilson@heanet.ie

Jean-Marc Uzé
Juniper Networks, Espace 21, 31 Place Ronde, 92986 Paris la Défense, France
e-mail: juze@juniper.net

Lars Fischer
NORDUnet A/S, Kastruplundgade 22, DK-2770 Kastrup, Denmark
e-mail: lars@nordu.net

Tomas de Miguel
RedIRIS, Edificio Bronce, Plaza de Manuel Gómez Moreno, s/n - 2a planta, 28020, Madrid, Spain
e-mail: tomas.demiguel@rediris.es

Abstract

This document provides the principles behind the MANTICORE project, whose main goal is to create an innovative service for the Network Operations Centre and end users that allows them to customize the configuration of their own dedicated IP physical and/or logical network through the use of a Web Services based Resource Management System. The motivation and the concept of this logical IP Network Service will be detailed and the software architecture of the first MANTICORE implementation will be explained (this implementation is based on the IaaS Framework, Infrastructure as a Service Framework). The conceptualization of the IP routing (internal and external) is important when providing logical IP networks; thus, the basic ideas will be explained in this paper. Future implementations will be able to extend the concept to layer 0, 1, 2 (by integrating MANTICORE with the Argia and Ether products) and/or layers above layer 3. Some of the MANTICORE future work, like the interoperability with the IPsphere Framework, will be carried out as part of the activities of the FEDERICA FP7 European Project.

Keywords

Logical IP networks, Infrastructure as a Service, IaaS Framework, UCLP, WebService, RPSL

Introduction

This project is intended to provide a Web Services based Resource Management System with offers a logical IP network. This system needs to control (logical) routers (initially based on Juniper architecture) devices. Beside a ready-made (logical) IP network, users will also be able to integrate logical routers into their own configurations and profit from the logical resources. To achieve this objective, the current version of IaaS Framework (Infrastructure as a Service Framework, based on UCLP) will be enhanced to support the logical routing feature and the XML API, starting from the Juniper system. In the future this IP (layer 3) Web Service will be integrate with existing or new Web Services for layer 0, 1 and 2 and if needed layers above 3.
The partners in this project (HEAnet, i2CAT, Juniper, NORDUnet and RedIRIS), will use their environment as part of defining the service and show cases.

The paper is divided in two parts. The first part presents and describes the definition of the user requirements of the logical IP network and an approach related to the most important part of an IP service:

the internal and external IP routing. The second part will look at the implementation aspects of this Web Service (WS).

The logical IP network

The use cases

The main motivation for looking into the possibility and feasibility of providing a virtualized IP service is the interest of network providers. Examples of network providers are the partners in this project, comprising National Research and Educational networks (NREN, such as HEAnet and RedIRIS), International networks (such as NORDUnet) and industry (such as i2CAT and Juniper).
The two main use cases for a logical IP network service are:

a) Provisioning (logical) IP network(s) for the service provider itself.
Here streamlining and standardizing the way IP networks are made is the main driving force. As soon as standardization is in place, it becomes easier for a network provider to provide logical IP networks to clients.

b) Provisioning of IP related resources towards users or other service providers
Users (be they humans or machines) might want to build their own IP network, and thus a service provider must be able to provide (logical) IP resources to individual users. This service might not be directly needed now, but it is foresee-able that it will be.

In general, similar use cases are pursued by lower layer services (like point to point, Bandwidth on Demand or light paths). The issues with these lower layer services are that they only cover part of the users needs (one to one connectivity) and special attention is needed to guarantee the routing integrity. Most users want, besides fast/predictable one-to-one connections, also many-to-many (IP) connectivity. By integrating lower layers in the concepts of the logical IP network we can achieve the best of both worlds.

To be able to support this environment, several items must be supported:

- (Logical) Router WS, a logical or physical device that has physical/logical ports and the ability to route traffic according to certain rules,
- Routing services, so that traffic in a router or network can be routed between internal entities (like other Router WS) or external entities (like users or external networks),
- Lower layer WS, which can provide connectivity at layer 0, 1 and 2 between (user/router) ports. For instance, optical patch panels (OOO), light paths, Ethernet or L2 MPLS VLL WS, etc.
- An IP network WS, which integrates/combines the above services.

Therefore, the idea is to combine the layer 0, 1, 2 and 3 and thus make a fully fledged IP service.

The routing services

One of the problems that have come into focus as a result of the wider availability of point to point links is that of ensuring routing integrity. As discussed at TNC2006 [1], the routing policies of NRENs and their clients have been built to match the effective hierarchy of connectivity that has existed until now. While we now have the facility to connect individual hosts, offices and campuses together, we may be unable to exchange traffic on layer 3 except in a very limited fashion without significant and intensive routing changes throughout that hierarchy. However, if one could provide a logical IP network with its own consistent routing policy, then users may be able to take advantage of this network for their own applications without adversely impacting the existing network on which they operate.

Every IP network is different, and providing a generic framework for the specification of any IP network is no small task. In order to provide a useful service, the logical networks that may be defined must be very flexible, with few or no restrictions on network topology, devices that may be connected, or routing protocols that may be supported.

However, when we are creating an entire logical network from whole cloth, there are some characteristics that we can rely upon, and then use to our advantage.

We treat a single logical network as a single entity for the purposes of routing, and draw a distinction between internal routing and external routing. In accordance with best practice, we use internal routing only to exchange information on the routers that make up the logical network. Because the information to be exchanged is minimized, the resulting routing table is small, and because we do not interact with "outside" devices, we have fewer concerns with the security and integrity of our internal routing. This lends itself very well to the use of well known IGPs such as OSPF and ISIS (and, since the protocol chosen only needs to run inside the logical network, we are free of many of the restrictions on that choice of protocol, and can choose it based on device support or any other particular characteristics the user might have in mind.)

Every other route, then, we see as external - not only other networks, but even those belonging to end hosts that must be distributed within the network and (in aggregate form) beyond. This is, perhaps, counterintuitive; it's reasonable to see end-hosts connected to a router as "within the network". Our contention is that, if a route is not necessary in order to connect *the routers* together, then it can be left outside the IGP. Since a number of router manufacturers now implement "indirect next-hop" for BGP, it is possible for BGP to converge just as soon as the underlying IGP converges. In any event, we must assume that BGP will converge *no quicker* than the IGP that glues it together.

This does require that all logical routers in the network be capable of speaking BGP, and one might fear that maintaining a large routing table on all routers might be unwieldy. It is true that if the user has a complex external routing policy which requires maintaining a full internet routing table, then all routers must maintain such a table, but best practice requires that all routers maintain a consistent routing table in any case, or else we risk encountering routing loops in some difficult to predict failure modes. If the user's routing policy is simpler - for example, a single upstream, or strict primary/backup upstreams - we plan, then, to keep the IGP no larger than the routers that speak BGP; to run BGP on all the routers in the logical network, even if there is a single external gateway; and to distribute end user subnets through BGP within our logical network. We believe this corresponds with best practice in operational networks today. Further, we can take advantage of such architecture in two ways.

Firstly, the configuration of the IGP is made very simple - in some cases, trivially simple. The user may still wish to specify parameters such as link metrics (either directly, or indirectly through, e.g., bandwidth and latency restrictions) but the interfaces on which the IGP must run are restricted to the loopbacks of each logical router, and the links which connect the logical routers to each other. This information can be deduced directly from the logical network layout specified by the user.

Secondly, if we are disciplined in using an BGP to distribute information on external (including end-user) routes, then we can take advantage of an already existing language for specifying routing information - RPSL.

RPSL is specified in the RFC 2622 [2] and extended in the RFC 4012 [3]. Its best known application in the European community is in the RIPE WHOIS database, where it is used to record the external routing policies of networks in the RIPE region. It is frequently used in production networks to automatically build route filters, and code has been published to generate router configurations from RPSL objects for various operating systems.

However, RPSLng is a much richer language than its most common uses might suggest, and can specify routing policy down to the level of detail of individual interfaces, including interactions with various IGPs and other protocols. It's even possible to specify some information on ISIS and OSPF with RPSLng.

This, together, makes the final service to the end user. Instead of requiring them (with extensive, specialist routing knowledge) to provision routers, links between routers, bring up various routing protocols, compose an appropriate routing policy, implement it, and document it with RPSL - we require only the layout of the network and their preferred routing policy. We then create the logical network in accordance with that specification, instead of requiring the layout and policy to be specified after the fact. In this way, we can provide a full logical network service to the user without requiring them to be unnecessarily familiar with the minutiae of logical links, routers and internal routing - and we can provide a working network, interfacing with other external networks, that is not subject to the routing integrity problems associated with connecting existing disparate networks over a point to point link.

First implementation of MANTICORE

The IaaS Framework

Virtualization is a current hot topic; a lot of research in virtualization related projects is being performed. However, the word virtualization may have different meanings to different people, so it is important to define what virtualization means in the context of the IaaS Framework and MANTICORE. We define virtualization as a technique which represents a physical device or substrate as a software entity (for instance an object, a web service or a web service resource). This technique has been widely applied to PCs; tools like VMWare, Xen or others provide several virtual machines on the same hardware. PC Virtualization, which was originally used locally as a means to reduce data centres cost by maximizing the server usage, has been extended by companies like Amazon or BlueLock that are among the firsts to offer a new type of services called Infrastructure as a Service (or IaaS), by renting hardware using proprietary solutions. IaaS is the equivalent of Software as a Service (SaaS) for hardware devices, where users do not buy hardware but instead pay for the use of it to a third party (which is the real hardware owner). During the duration of the service, the user owns and controls the infrastructure as if he was the owner; this business model is targeted towards long term resource usage compared to on-demand services.

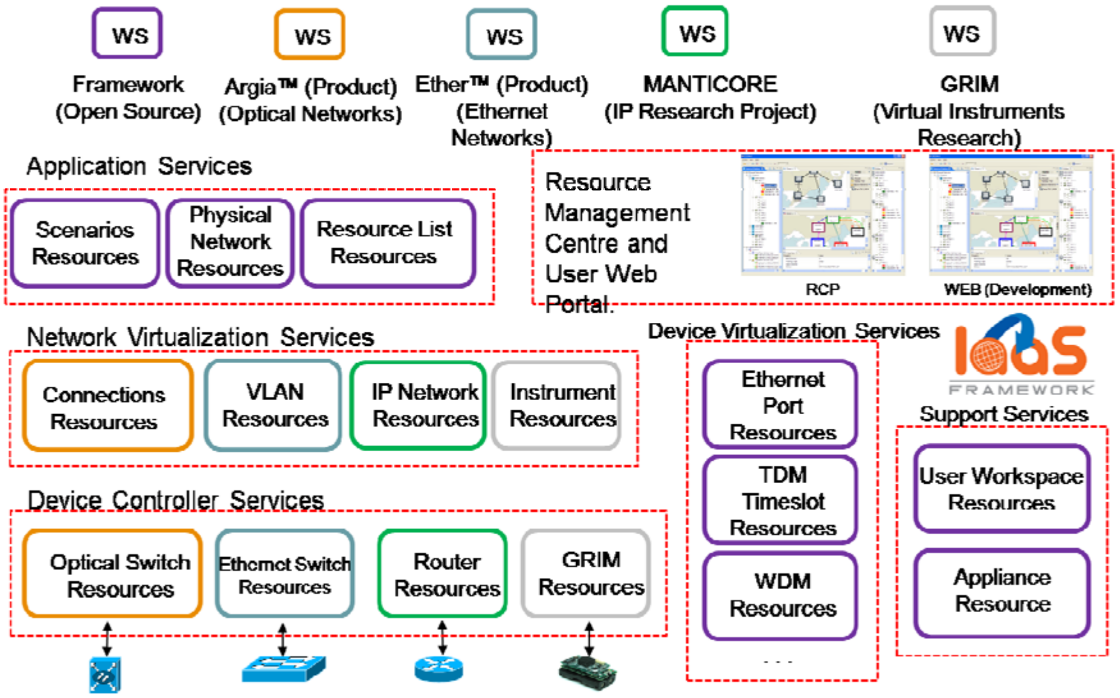

Figure 1. Architecture of the IaaS Framework and the framework based products and research projects

The IaaS Framework [4] is a generalized approach to the outcome of years of research under the UCLP [5] research programs funded by CANARIE. Beginning in 2001, two UCLP research projects were put in place to provide a virtualization solution for optical networks: UCLP initial goal was to provide end to end paths across domains; UCLPv2 goals were to create reusable and configurable network blocks. UCLPv2 concepts have evolved into many different Physical to Virtual (P2V) products and research projects that are built on the IaaS Framework: Argia [6] is the product for optical networks (TDM, WDM and Fiber technology), Ether will be the product for Ethernet and MPLS networks, MANTICORE is the research project for logical IP networks and, finally, the GRIM project performs research on applying virtualization to instruments and sensors.

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.

Figure 1 shows an overview of the architecture of the IaaS Framework and the framework based products and research projects. Device Controller Services are a group of services that act both as physical device controllers, i.e. they communicate with the physical device using the required protocol, and as virtualization factories, that is, they provide the “virtualize” operation that causes a new resource representing a part of the physical device to be created (for instance a group of TDM channels of a port in a SONET switch or an Ethernet interface of a router). The resources created as a result of the “virtualize” operation belong to the Device Virtualization Services group. These services represent a part of a physical device: Ethernet Port Resources represent Ethernet interfaces, TDM Timeslot Resources represent groups of channels in TDM interfaces, WDM Resources represent wavelengths in WDM interfaces and so on. These resources are the ones being exchanged by organizations to provide the infrastructure as a service. For instance, if organization A owns a SONET switch with 8 TDM interfaces and 16 Ethernet ports, the administrator of organization A could virtualize 4 TDM interfaces and 8 Ethernet ports and give them to organization B, so that organization B could control these ports as if it was the real owner. Furthermore, organization A could virtualize the other 4 TDM interfaces and 8 Ethernet ports and give them to organization C, so that both organization B and C would control parts of the same optical switch as if they were the real owners. Network Virtualization Services are the services that provide an end-user over the virtual infrastructure (or infrastructure as a service). Examples of network virtualization services are Argia’s optical connection service, which allows a user to perform point to point connections over TDM, WDM or Fibre Resources; or MANTICORE’s IP Network Service, that will be explained later in this paper.

Resource Lists Services provide the means of exchanging resources between organizations. When organization A wants to give permission to organization B to access some of organization’s A resources, organization A creates a resource list populated with device virtualization resources that represent all the physical infrastructure that organization A can access, and sends the resource list to organization B. When organization B receives the resource list, it can assign the device virtualization resources it has received to one or more of the network virtualization services organization B has deployed (for instance, organization B could say: these 4 ethernet ports can be used by the IP Network service and these 8 TDM ports can be used by the Optical Connection Service). This resource exchange process is recursive, as shown in Figure 2, meaning that organization B could create another resource list and give access to some of the resources that organization B temporarily owns to another organization. A future improvement is to allow an organization to export or publish a resource list to a well-known resource broker site, such as http://www.vinfrastructures.com, so that interested organizations can browse these sites and get the infrastructure resources they need.

Organizations and its users are managed by the User Workspace Service. Each user in an organization has its user account and its associated credentials that will be used to interact with any of the services. Users can have three different user roles:

- Physical Infrastructure Administrator. Owner of a physical infrastructure. This type of user can virtualize the physical resources he owns and assign permissions for other users to control them (by creating and exporting resource lists).
- Virtual Infrastructure Administrator. This type of user gets resources from one or more Physical Infrastructure Administrators or other Virtual Infrastructure Administrators. He can also assign the resources he can control to different network virtualization services.
- End User. This type of user is the typical "dumb" end user that just wants to use a service (like an end-to-end connection service or an IP Network service).

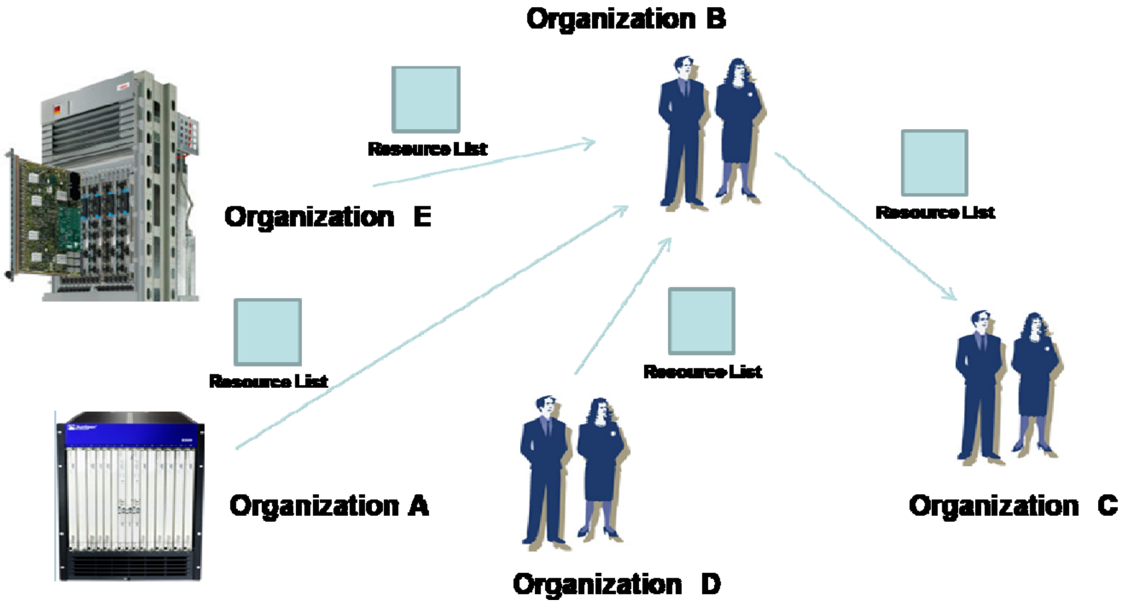

Figure 2. The resource trading model enabled by the IaaS Framework

MANTICORE Software Architecture

As explained in the previous section, MANTICORE's software architecture, depicted in Figure 3, is based on the IaaS Framework software. MANTICORE particular software modules are the Router WS (a device controller service) and the IP Network WS (a Network Virtualization service).

The Router WS is a WSRF [7] based web service that can interact with one or more routing physical devices. The routing device state (inventory, including physical and logical interfaces, logical routers, alarms and configuration) is exposed as a WS Resource, so that clients of the Router WS can access the state of the physical routing device by querying the resource properties. The Router WS interface provides a series of high level operations that encapsulate the routing device functionality. For instance, using the "invoke" operation and passing the appropriated operation identifier and parameters, a Router WS client could configure the network mask of an interface, create/delete logical interfaces, insert/remove static routes, configure dynamic routing protocols such as RIP, OSPF and BGP, and so on. It is interesting to note that the parameters the operations have to work with cannot be for any particular routing device, but they ideally must work for all the router devices. As an example, the operation to configure OSPF in a router interface must be able to work with the parameters that a Juniper router would need as well as with the parameters that a Cisco router would need.

The Router WS 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 trough the NetConf API [8]. 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.

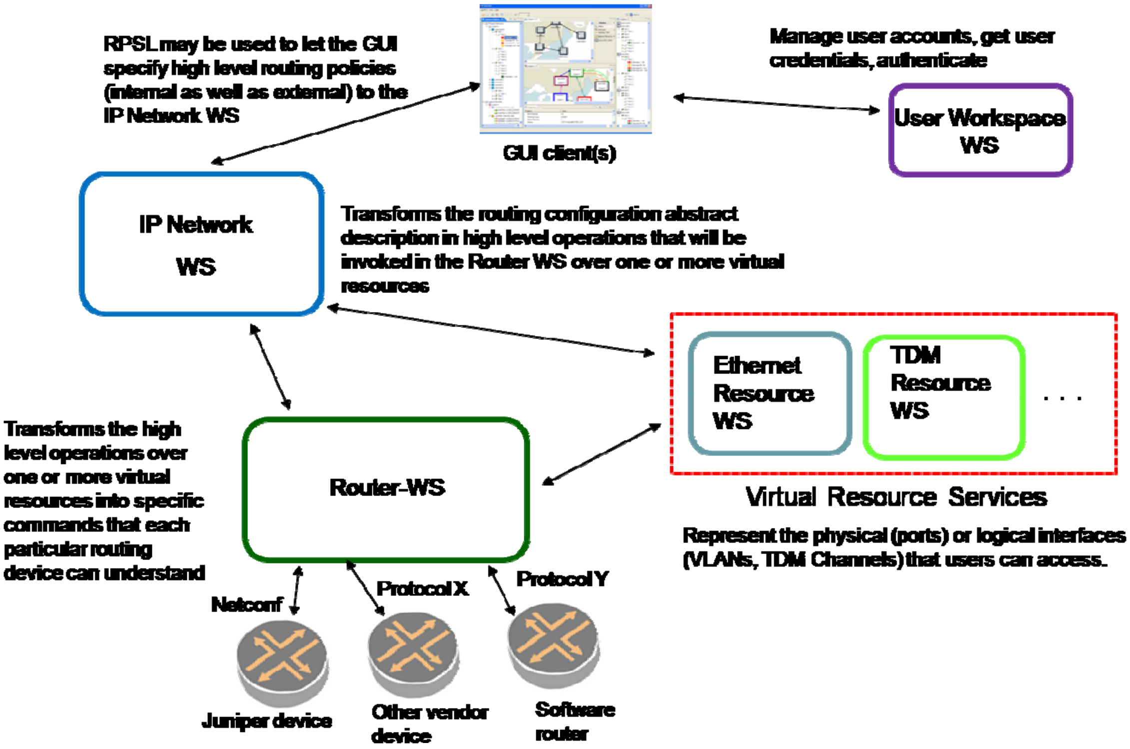

Figure 3. MANTICORE Software Architecture

Physical Infrastructure Administrators call the Router WS “virtualize” operation to create Ethernet Resource WS that represent Ethernet physical or logical interfaces of the router. These Ethernet Resource WS can be later added to Resource Lists and exported to other organizations so that their users can control the physical infrastructure.

The IP Network WS is also a WSRF web service that manages one or more IP Network Resources. Each IP Network has pointers to 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 provides the following capabilities to its clients:

- General configuration of the router interfaces. The user can configure the IP address, the NETMASK, 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 a 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).

A simple yet expressive and extensible data structure or language has to be found to be able to model the configuration data to be sent to the IP Network WS. This data is mainly IGP configurations or routing policy specifications, and it must remain device independent. As explained in the “Routing Services” section, the RPSL language has been analyzed to see if it is suitable for modelling routing policies as well as IGP configurations in a vendor independent fashion. Our investigation has concluded that it is feasible to extend RPSL to represent at least simple IGP configurations; therefore it is suitable to be the modelling language used by the IP Network WS. However, an RPSL java module has not been implemented yet and integrated within the IP Network WS; so far a proprietary ad-hoc simple language is used in the current prototype.

The IP Network WS lifecycle is the following. Once it is created, the IP Network WS is populated with pointers to a set of Ethernet Resource WS, which represent the interfaces it can access. IP Network WS clients invoke its operations, sending the configuration data in an abstract language with the request, and the IP Network WS validates the request, infers what resources have to be configured, checks that it can access these resources, and creates the required requests to the Router WSs affected by the client configuration. If the operations are successful the IP Network WS updates the Ethernet Resource WS resource properties. If the user wants to destroy the IP Network, a destroy call is issued to the IP Network WS, and the resource representing the particular IP Network is removed from the system.

Future Work

MANTICORE first implementation has achieved its goal: to serve as a proof of concept to demonstrate the new capabilities enabled by the IP Network Service, but more work is needed to achieve the next step: improving the software so that it is sufficiently robust to be deployed in a production environment and it can be integrated with the other tools that the network provider may use to deliver other services, such as point to point circuits. To achieve this goal two activities must be completed: one is to improve MANTICORE’s implementation, especially the error handling modules, the support for more devices and the software overall performance; the second one is to design and implement an integration plan with Argia (the IaaS Framework based product that provides a P2V solution for optical networks) and with Ether (the IaaS Framework solution for delivering P2V to Ethernet and MPLS networks) once it is implemented. By using MANTICORE, Ether and Argia in conjunction, the service provider could offer its entire network infrastructure from the optical layer up to the IP layers as IaaS.

Another topic of interest is to improve the functionality of the IP Network WS. Autonomous Systems exchange agreements with their neighbours in order to determine how the AS routes traffic that enters or exits its infrastructure. Uncoordinated routing policy decisions between AS’s can adversely impact Internet connectivity. In the absence of policy coordination, it is possible for different AS’s to establish conflicting routing polices. In this field, RPSL is a Routing Policy Specification Languages. It allows a network operator to be able to specify routing policies at various levels in the Internet hierarchy; for example at the Autonomous System (AS) level. At the same time, policies can be specified with sufficient detail in RPSL so that low level router configurations can be generated from them. DML (Domain Modelling Language) is a simple language, yet very powerful in terms of expressiveness. SSFNet uses DML to describe the network models. DML is not thought to define routing policies. However, by means of the definition of all the network elements, it is capable of defining the policies. The problem is its complexity that implies the definition of all the elements in a complex network.

RPSL is a well suited language for the definition of routing policies between ASs. The routing policy specified for intra-domain routing is not as necessary as Inter-domain routing, however the RPSL capacity in the intra-routing policy definition will continue to be explored. Therefore, some extensions are required for intra-autonomous systems routing policies definition. Because RPSL can be extended, new routing protocols and new features can be added. For example, when some non-shortest path routes are preferred, a deeper RPSL analyses is needed.

On the other hand, routing in logical instances of a physical router means that physical interfaces are shared. Therefore, the bandwidth of each link is shared among several logical routers. According to this, a fair criteria is needed in order to allocate bandwidth; doubly so, when the bottleneck is elsewhere in the network.

Activities under the FEDERICA FP7 Project

FEDERICA [9] is a European project of the 7th Framework Program in the area of "Capacities and Research Infrastructures". The FEDERICA project will create a European wide "technology agnostic" infrastructure made of Gigabit circuits, transmission equipment and computing nodes capable of virtualization to host experimental activities on new Internet architectures and protocols.

MANTICORE will continue part of its activities inside the FEDERICA project. The work to be carried out inside FEDERICA includes adding support for more routing devices, including physical routers from other vendors as well as software routers, and developing new software modules to be able to interoperate with the IPsphere Framework [10], [11]. The IPsphere Forum –a non-profit association of vendors, network operators and content providers- has developed the IPsphere Framework specification, an open multi-stakeholder web-services framework for the rapid creation and automated deployment of IP based services. This framework allows service providers (network operators, content providers) to publish the services they offer into some well known registries. A special type of user called Administrative Owner (similar to the IaaS Framework APN Administrator), can compose multiple of the services offered by the service providers into an end to end solution, and offer it to the end user. The interoperability implementation must allow Administrative Owners of IPsphere to compose resources coming from MANTICORE (e.g. Logical routers or IP Networks) into an IPsphere Service.

Acknowledgements

The authors would like to deeply thank Jordi Ferrer, Carlos Baez, Juan Felipe Botero, Josep Pons and Alejandro Berna (all from the Universitat Politècnica de Catalunya, Spain) for their contributions to this work.

References

[1] Dave Wilson, "Routing Integrity in a world of Bandwidth on Demand", TNC 2006. Available online at http://www.terena.org/events/tnc2006/programme/presentations/show.php?pres_id=242
[2] RFC 2622, "Routing Policy Specification Language". Available online at ftp://ftp.rfc-editor.org/in-notes/rfc2622.txt
[3] RFC 4012, "Routing Policy Specification Language next generation". Available online at ftp://ftp.rfc-editor.org/in-notes/rfc4012.txt
[4] The IaaS Framework website, available online at http://www.iaasframework.com
[5] Eduard Grasa, Sergi Figuerola, Albert López, Gabriel Junyent, Michel Savoie, Bill St. Arnaud and Mathieu Lemay, "Articulated Private networks in UCLP", Internet Research: Electronic Networking Applications and Policy, Vol. 17 No. 5, 2007, pages 535-545.
[6] Inocybe Technologies Inc. website, available online at http://www.inocybe.ca
[7] OASIS, "Web Services Resource Framework – v1.2Primer" Available online at http://docs.oasis-open.org/wsrf/wsrf-primer-1.2-primer-cd-02.pdf
[8]NETCONF Configuration Protocol, RFC 4741, December 2006. Available online at: http://www.ietf.org/rfc/rfc4741.txt
[9] FEDERICA Project homepage. Available online at: http://www.fp7-federica.eu/
[10] IPsphere Forum, "IPsphere Framework Technical Specification, Release 1", June 2007. Available online at: http://www.ipsphereforum.org/library/specifications/R1.html
[11] Jean-Marc Uzé, "IPSphere Forum: Status on Technical Specifications", TNC 2007. Available online at http://tnc2007.terena.org/programme/presentations/show.php?pres_id=94

Abbreviations

API	Application Programming Interface
BGP	Border Gateway Protocol
EGP	External Gateway Protocol (e.g., BGP)
FEDERICA	Federated E-infrastructure Dedicated to European Researchers Innovating in Computing network Architectures
IaaS	Infrastructure as a Service
iBGP	Internal BGP
IGP	Internal Gateway Protocol (e.g. ISIS, OSPF)
ISIS	Intermediate System to Intermediate System
IP	IPv4 or IPv6 protocol
MPLS	Multi Protocol Label Switching
MANTICORE	Making APN Network Topologies on Internet COREs
NREN	national Research and Education Network
OOO	Optical Optical Optical switches
OSPF	Open Shortest Path First
RIP	Routing Information Protocol
RIPE	Réseaux IP Européens
RPSL	Routing Policy Specification Language [RFC2622]
P2V	Physical to Virtual
RPSL	Routing Policy Specification Language [RFC2622]
UCLP	User Controlled LightPaths
VLL	Virtual Leased Line
WS	WebService
WSRF	Web Services Resource Framework
XML	Extensible Markup Language

Vitae

Eduard Grasa graduated in Telecommunication Engineering from the Technical University of Catalonia in 2004. In 2003, he joined the Optical Communications Group (GCO), where he wrote his master thesis on UCLP. In June 2004, he joined the i2cat foundation as a member of the UCLP project and is a PhD candidate of the GCO (UPC). He has participated in the enhancement of the UCLPv1.x software and is one of the original developers of the UCLPv2 software. He is currently involved in the development of the IaaS Framework initiative and the Argia product.

Dr. Xavier Hesselbach is an Associate Professor at the ETSETB, in Barcelona, Spain. He received the M.S. degree in Telecommunications Engineering in 1994 and the PhD. degree in 1999, from the Technical University of Catalonia (UPC). In 1993, he joined the Design, modelling and evaluation of broadband networks group in the Telematics Engineering Department of the UPC. He has been involved in several national and international projects. His research interests address MPLS traffic control and GMPLS networks, QoS and traffic grooming in IP/MPLS over optical networks. He is author of more than 40 national and international publications in conferences and journals. He is co-author of 4 books. Since 1992, he is member of the IEEE. In 1994 he received the award from the COIT/AEIT of Spain for the best Master Thesis on Networks and Telecommunication Services.

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 Ramón 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. He is currently acting as Work Package leader of the PHOSPHORUS project (EC FP6) and Activity leader of the FEDERICA project (EC FP7).

Victor Reijs has studied at the University of Twente in the Netherlands, worked for KPN Telecom Research and SURFnet, since 1999 working for HEAnet. He was involved in CLNS/TUBA (one of the earlier alternatives for IPv6). Experience was gained with X.25 and ATM in a national and international environment. His last activity at SURFnet was the tender for SURFnet5 (a step towards optical networking). Emigrating to Ireland, he is managing the Network Development department of HEAnet and is actively involved in international activites such as GN2, TF-NGN and Grids as well as on (optical) networking, point-to-point links and monitoring.

Dave Wilson studied in University College Dublin, and has worked on HEAnet since 1996, joining HEAnet Ltd on its establishment in 1997. Dave is a Senior Network Engineer working for the Network Development team. His role of Network Planning involves him in subjects such as IPv6, RIPE and GEANT. Last year Dave was elected by RIPE members to the ICANN Address Supporting Organisation Address Council, which has a role in global number policy. Dave spoke at TNC 2006 on the subject of routing integrity with Bandwidth on Demand, and is currently working on the rollout of HEAnet's new IP networking platform.

Jean-Marc Uzé has been a consultant at Juniper Networks since 2001, focusing on Research, Education and Government Networks and Institutions. He spent 4 years at GIP Renater (the French Academic Research Network). He directed the Renater 2 Project, the new generation National Research Network of France and managed several projects such as TEN-34, TEN-155 and US connectivity. He also coordinated the MPLS activities of the European technical Task Force TF-TEN and TF-TANT. Jean-Marc has a Master of Science in Network Engineering, and started his career as head of the Data-processing centre of INRA, the French Agronomic Research Institute in Versailles.

Lars Fischer has been the CTO of NORDUnet since 2004 and is active in international collaborations such as GN2, GLIF, TERENA and works among other things with optical and IP networking and grid computing. He has worked with research infrastructure for more than 20 years, starting at Aalborg Universirty where he helped introduce IP networking and support an NREN in the mid 80's. Lars holds a M.Sc.E.E. from Aalborg University, Denmark. Before joining NORDUnet he worked Tele-2 and COLT Telecom.

Tomas de Miguel is Director of the Spanish National Research Network (RedIRIS) and doctor in telecommunication engineering by the Technical University of Madrid (UPM). He has been associate professor at UPM, where he imparted informatics and communications courses and developed his research activity and technical management in the Engineering and Telematic Systems Department (DIT), working on formal specification of protocols and technologies for the new generation of Internet, especially with IPv6 protocol, advanced collaborations technologies and services federation. Prior to RedIRIS, Tomás was responsible of informatics and communication services at E.T.S.I Telecomunicación UPM, providing new high speed networking capabilities to facilitate new Internet services introduction.