Service Oriented framework for IPv4 to IPv6 Transformation

phd. student Nikolay Milovanov, msc. Vessela Georgieva, prof. Antoni Slavinski

New Bulgarian University, Telecommunications Department, Montevideo 21,
1618 Sofia, Bulgaria,

{nmilovanov, aslavinski}@nbu.bg
{vgeo}@niau.org

Abstract. The target of 4 to 6 Transformation is a research field and business process related to the network infrastructures migration from IPv4 to IPv6. Currently only about 8.6% of the IPv4 address space is free therefore that process could be considered inevitable. The migration to the new protocol could be pretty expensive for the network operators and extremely disruptive for the end customers. The Project aims to deliver an open framework capable of handling the migration process. The framework consists of three major components – Business Transformation Logic (BTL), Network Inventory and Service Transformation. Each of the components extends the Fulfillment, Assurance and Billing (FAB) Model and could be easily integrated with the current Operation Support Systems (OSS).

Keywords: IPv4 to IPv6 network transformation, OSS, FAB model, Service Oriented Framework

1 Introduction

This paper describes the architecture proposal for an open service oriented IPv4 to IPv6 network transformation framework that will extend the Fulfillment part of the current OSS solutions. The prospect is developed under the 4to6TRANS project initiative. The project aims to deliver open source tools for IPv4 to IPv6 network migration to all kinds of network players throughout Europe.

Currently IPv6 is matured enough and is already widely supported by the network industry vendors and software manufacturers. Most of the Operation Systems, Browsers, Email Clients, Web and DNS Servers already support IPv6. The share of free IPv4 address space is getting smaller and it is expected network providers to migrate their current services and subscribers to the new protocol [1]. Nevertheless, still a real migration to the new protocol does not occur. It will be costly and extremely disruptive, which is the major factor impeding its real start. Another reason is that currently, there are no Operation Support Systems tools, software, commercial and open platform able to handle similar transition. The reason for that is the fact that

IPv6 as IPv4 is just a technology. It provides new opportunities and features but does not directly provide new services that usually are the market driver. It is quite difficult in the current market situation somebody to invest significantly in such migration. The operators do not want to migrate to IPv6 because it will be costly, current network is already too complex and their current OSS won't able to handle. From the other perspective OSS vendors does not want to invest significantly and to rework their products to support the new protocol without a direct purchase orders by the operators. So in the end we have a chicken and egg situation.

With 4TO6TRANS initiative the project authors would like to push a bit the industry and to show that such migration is possible with the right knowledge, tools and approach.

2 4TO6TRANS project concepts and objectives

Facing that problem, realizing that similar network migration task being extremely complex the project authors propose a novel 4to6TRANS framework to be created, having the power, flexibility and ability to model the current services, to speak with the network devices and to follow certain business logic during the transformation process.

The framework architecture consists of several Application Programmable Interfaces build around a relational database and following the principles of Service Oriented Architecture. Since such a job is quite complex and requires a lot of effort and resources the framework and its tools will be realized as an Open Source Project under the 4to6trans initiative.

The framework is supposed to give the possibility to the Network Operators to perform the transition in controlled and automated fashion. The final project output will be of great help to a big number of Internet Service Providers, Cable, Telecoms and Mobile operators to optimize the transition process and to reduce the loss of service periods in their networks. From the other perspective the end users won't experience a significant loss of service during that service transition periods.

The project focuses on service transformation of ISP residential clients, but also on Inter ISP Border Gateway Protocol (BGP) peering and corporate client services. The most common current ISP architecture is based on Multi Protocol Label Switching (MPLS) backbone throughout service provider infrastructure and several service clouds build around it. The framework will be robust and flexible enough to handle easily the communication with great diversity of network devices as routers, switches, radius/diameter servers, Policy Control Points, xDSL modems, WiMax base stations and IPTV platforms.

The residential services might be:

- Internet Access incorporated with Hosting, Email and AntiX
- IP Telephony
- IPTV

The business services might be:

- Internet Access (Email, AntiX, Hosting)

- Data Virtual Private Network (VPN) between the company locations (MPLS L2 or L3 VPN)
- IP Telephony Solution
- Site to Site encryption
- Remote Access

As a final result the framework shall be able to transform the described IPv4 based services and platform to an IPv6 such if IPv6 is supported on that platform. If the protocol is not supported a transition methodology shall be proposed and configured on the devices of the network operator.

For the purpose will be specified and created several open application programmable interfaces (API) able to model the transformation process and several GUI tools to be used by the OSS designers and OSS end users for an easy API manipulation.

The first stage of the project is the specification of the framework's API. The API's are grouped in three major groups.

- Business Transform Logic (BTL)
- Network Inventory
- Service Transformation

The output of that phase is:

- Detailed framework architecture specification
- Network Inventory APIs software specification
- BTL APIs software specification
- Service Transformation APIs software specification

The second phase is APIs creation. Depending of the specification the APIs will be written in different software technologies. For example:

- BTL might be written using Open Source Service Oriented Architecture technologies.
- Network Inventory might be created around a relational database using the procedural language built in the database itself.
- Service Transformation able to model the communication with the network devices.

The outputs of that phase will be:

- Network Inventory APIs source code
- BTL APIs software source code
- Service Transformation APIs source code

The third part of the project is on testing and integrating the framework with real network devices. Several network architectures will be simulated in a lab environment. One of the project objectives is to test the API's against different network vendors. Therefore several equipment vendors will be contacted and framework interoperability tests will take place.

- The goals that shall be achieved in this phase are:
- Perform Inter vendor Network Discovery
- Perform Inter vendor Device/Service Upload
- Model certain service business logic through workflow creation

Transform IPv4 services and subscribers to an IPv6 following certain business logic.
The outputs will be:

- Network Discovery documentation
- Network Upload documentation
- IPv4 to IPv6 Service Transformation Prove of Concept Tests documentation

3 State of the art in the field of the current OSS

3.1 Current standards and specifications

The 4to6trans project follows the general concepts of the OSS (Operations Support Systems) based on the ITU-T recommendations and best practices proposed by the TMforum. Fig. 1 presents the FAB (Fulfillment, Assurance and Billing) model part of the E-tom model architecture [11], [12].

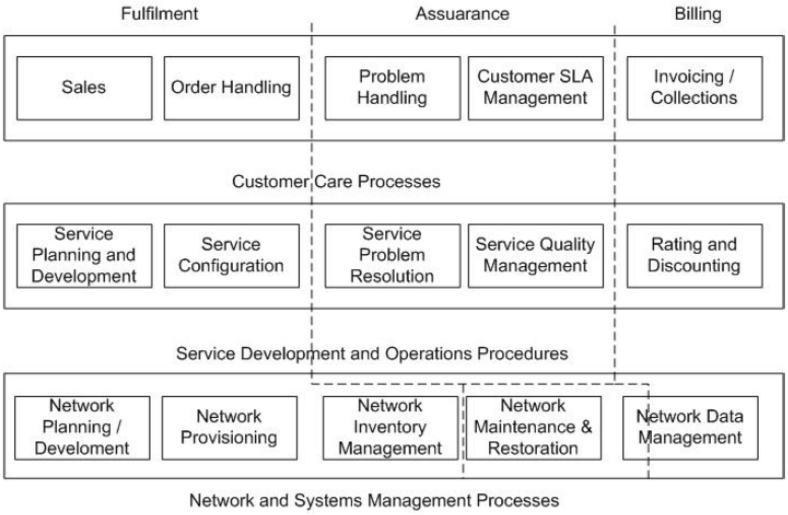
Fulfilment
Assuarance
Billing
Sales
Order Handling
Problem Handling
Customer SLA Management
Invoicing / Collections
Customer Care Processes
Service Planning and Development
Service Configuration
Service Problem Resolution
Service Quality Management
Rating and Discounting
Service Development and Operations Procedures
Network Planning / Develoment
Network Provisioning
Network Inventory Management
Network Maintenance & Restoration
Network Data Management
Network and Systems Management Processes

Fig. 1. Fulfillment assurance and billing model

The 4 to 6 transformation framework extends the Fulfillment part of the FAB model. The BTL will be added to the current “Service Planning and Development” and “Service Configuration’ components. The current “Network Inventory” will be extended with the objects needed for the service transformation. The Network Provisioning will be extended with the “Service Transformation API”.

3.2 Other similar solutions

Currently there are many OSS systems following the recommendations of the TM forum and delivering some of the components of the fulfillment branch of the FAB model. These platforms could be classified as platforms produced by Network Equipment Vendors that support only their equipment and independent OSS vendors that support the equipment of many vendors. Among the first group are companies like Cisco, Eriksson, Nokia, Siemens, Juniper, Huawei and many other network and telecom equipment vendors. In the second group are companies like Amdocs, Telcordia, Comptel, Sigma Systems and many others.

From the perspective of 4 to 6 service and device configuration transformation, considering the fact that most of the Network and Telecom operators have services realized on the equipment of many vendors we consider that the first group of OSS systems could not properly handle the process. Some of the companies' that are part of the second group OSS software vendors are producing solutions that could be considered as a state of the art in the OSS business field. Unfortunately neither of those companies targets seriously the IPv4 to IPv6 transformation process. Those platforms neither have an automated device upload, nor an automated discovery process. Usually once the network resources are entered into such system they could not be easily changed.

Those solutions may have network inventories, workflow managers and service activations but neither of those could handle the complexity of such complex service transformation.

4 Progress Beyond the state of Art

The 4to6TRANS project aims to deliver a software framework and practical guidelines for smooth and controlled migration to IPv6 with minimal loss of service. All Internet Service Provider, Telecom, Cable or Mobile operator shall sooner or later migrate to IPv6. Each one of those organizations will benefit greatly from the proposed service transformation platform. Since such migration will directly reflect to each and every customer of those organizations we may conclude that most of the population of the European Union using Internet Services, Mobile Services or data connectivity will benefit indirectly from the project outputs. Currently there is no product or platform able to handle such a complex migration task. This section will identify the progress beyond the state of art that will be achieved through the successful completion of the 4to6TRANS project.

4.1 Solution Architecture

The solution architecture consists of several layers (fig.2). The upper layer provides an interface for northbound integration with external systems and interfaces. That is the layer from which the framework receives the Service transformation Orders. On the next layer reside two of the main framework components the BTL and the Network/Service and Subscriber Inventory. The second layer components are able to communicate between each other through an open APIs and also with the upper and the lower layers. In that layer resides all polices and rules for service transformation. It contains the logic behind the transformation process. It also contains all the data for the network subject of transformation. The lowest layer main purpose is to process the request from the upper layer and to the network. It shall handle different communication protocols with different kinds of network elements.

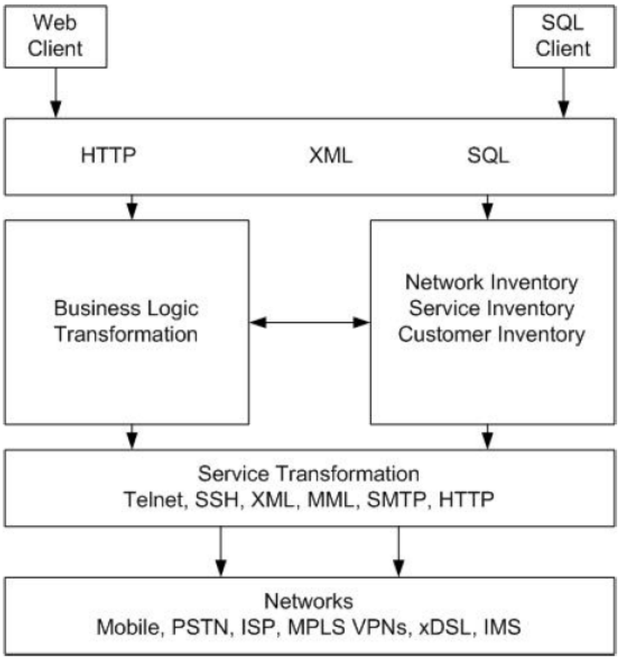

Fig. 2. Solution Architecture

4.2 Device Discovery

One of the main goals of the project is to have an automated upload of the device configurations into the network inventory. Such process will save large amounts of time to the network providers prior to the transformation phase. Therefore network discovery algorithm has to be implemented into the system. As an initial input prior to the discovery start into the system has to be entered an initial IP address and a stop rule. For example if MPLS P router is discovered the discovery process shall stop. Then it has to analyze the outputs from certain commands and SNMP MIBs in order to determine the device neighbors. Once the neighbors are identified the system will login to them, will identify their neighbors and continue the discovery process. Once certain device is discovered the Network Inventory will perform a full upload of that device. The discovery algorithm has to stop in case the device is already discovered or in case the stop rule is resolved.

4.3 Device Upload

Devices upload aims to model router's logical service model into the network inventory. Network inventory has to be populated only with a subset of the device configuration that will be needed by the BTL for a successful transformation. For the purpose the transformation platform has to communicate with the device through a set of standard or proprietary protocols. Such protocols could be CLI (Telnet, SSH), SNMP or MML. Clearly, there will be a preference for SNMP MIB browsing rather then the rest of the protocols.

The upload might be triggered for a group of devices (e.g. upon a network discovery completion) or an individual one (e.g. manual request) the process will systematically interrogate the individual devices to obtain the necessary data to achieve the representation in Inventory.

There exist a number of discrepancies in the way the devices return the requested information even across different models/platforms belonging to the same manufacturer. Therefore, the information gathering process will need to be resilient enough to deal with different response formats.

Once the information is gathered from the network device, it will be processed and the relevant data modeled into Inventory. Figure 3 displays a high-level view of the process.

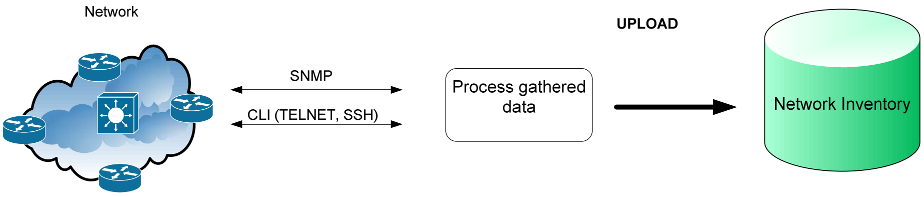

Fig. 3. Upload Process Overview

4.4 Network/Service Inventory

Once the upload process is finished the network inventory has to contain a logical data model of the network and its devices. The inventory has to contain all the information needed to the BTL in order to execute a certain transformation workflow. A simple logical model may have hierarchical structure similar to the one on fig.4.

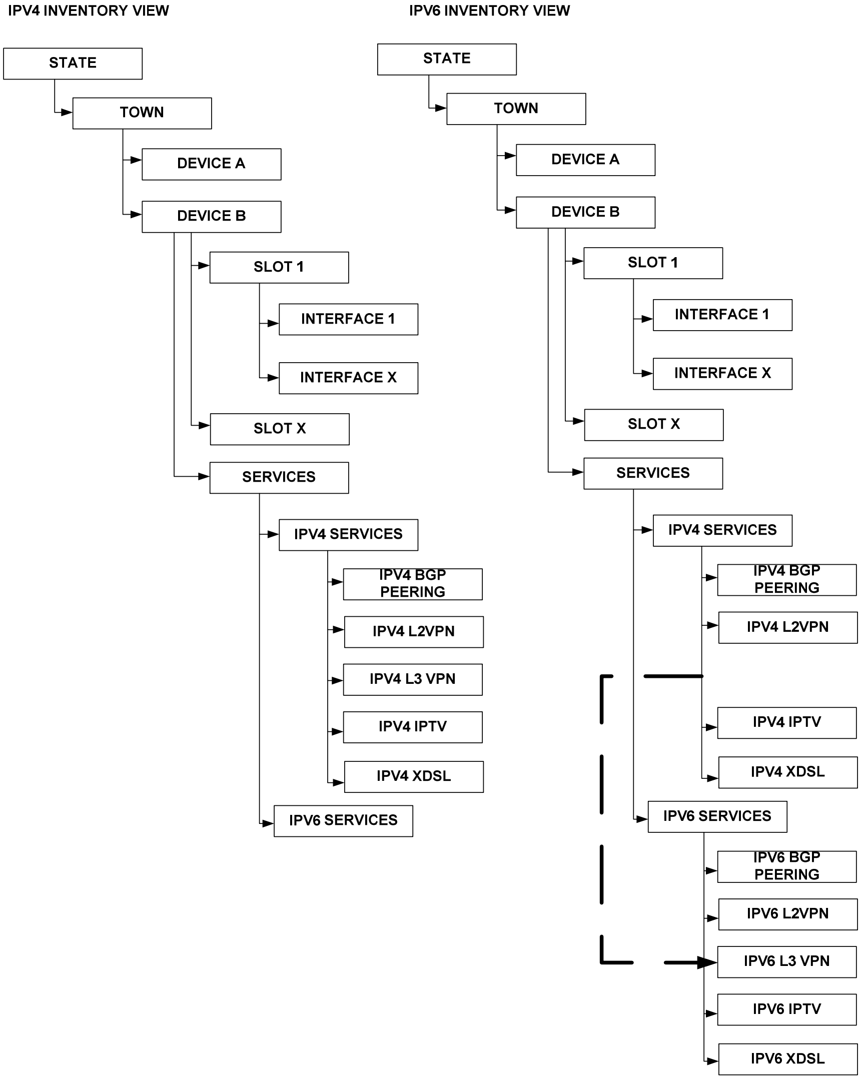

Fig. 4. In the left is presented an IPv4 Inventory and in the right Network/Service Inventory after a successful transformation to IPv6

4.5 Service Automation

Once the inventory is populated with the logical model of the network its devices and services. Then that information shall be used for as input parameters for the transformation logic. Service automation in the 4TO6TRANS will be achieved through the Business Transform Logic API. That part of the solution has to transform the IPv4 services into IPV6. The first main goal of the Business Transform Logic is to be able to model and execute a certain transformation algorithm. High level overview of an example transformation algorithm (workflow) that shall be modeled through this GUI is presented on Fig 5.

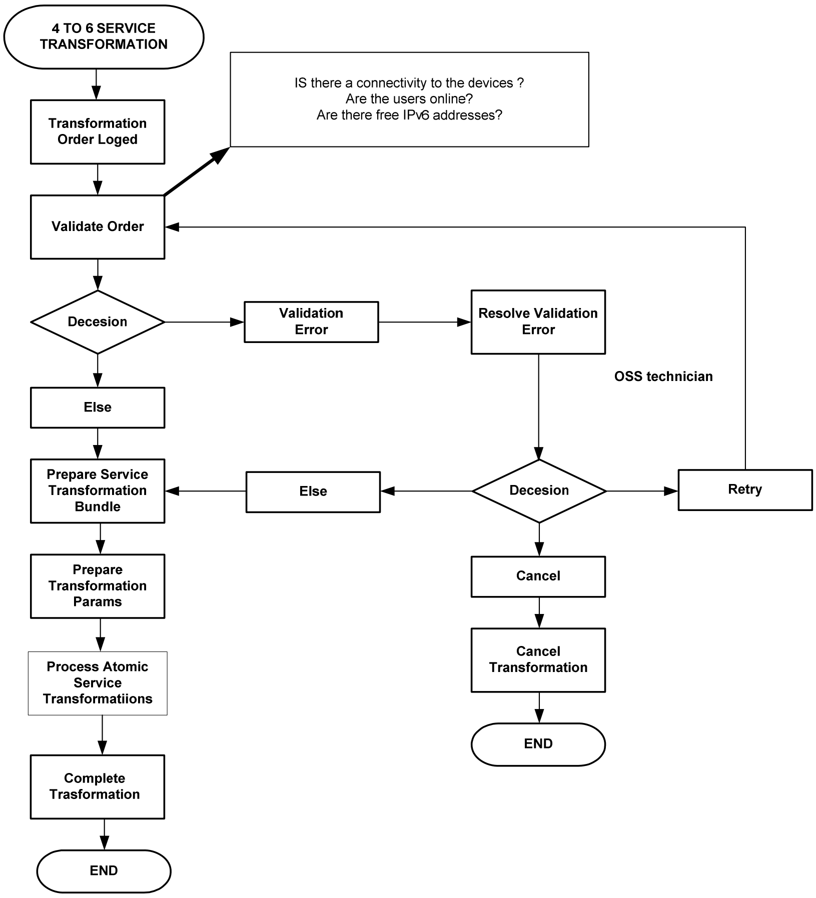
4 TO 6 SERVICE TRANSFORMATION
IS there a connectivity to the devices ?
Are the users online?
Are there free IPv6 addresses?
Transformation Order Loged
Validate Order
Decesion
Validation Error
Resolve Validation Error
OSS technician
Else
Prepare Service Transformation Bundle
Else
Decesion
Retry
Prepare Transformation Params
Cancel
Cancel Transformation
Process Atomic Service Transformatiions
END
Complete Trasformation
END

Fig.5. An Example for Service Transformation Process/Algorithm

The algorithm consists of a single workflow, several tasks and decision operators. Let's consider as an example that this workflow is transforming a business IPv4 MPLS L3 VPN service to IPv6 such. In this case the task breakdown will be:

1. Start – Formal start task.
2. Log Transformation Order
    2.1. Input (Customer Name, Customer Locations, Virtual Routing and Forwarding (VRF) Instance name)

2.2. Transformation time frame (e.g the customer has agreed with the operator his service to be transferred to the new protocol in certain time frame). For example during the weekend when customer office locations are empty.

3. Validation
   - 3.1. Customer exists.
   - 3.2. Billing verification - Customer has paid his service for the last 3 months.
   - 3.3. Data up to date - Customer data into the Inventory is up to date (if not a new upload of the customer devices have to be triggered).
   - 3.4. IPv6 Support - Customer equipment supports IPv6
4. Resolve Validation Error
   - 4.1. For example customer equipment in certain location does not support IPv6
   - 4.2. An email notification will be triggered to the NOC technician.
   - 4.3. NOC technician might realize that customer equipment software version does not support IPv6 and might trigger a manual software upgrade.
   - 4.4. Once the update is done the device will be uploaded again into the Inventory and the system might proceed to the next task.
   - 4.5. If the upgrade is unsuccessful the Technician might cancel the transformation.
5. Prepare Service Transformation Bundle
   - 5.1. Since some of the customers might have more then one VRF on this step the system shall decide based on the input parameters which VRF shall be transformed.
6. Prepare Transformation Parameters
   - 6.1. Once the bundle is determined the algorithm shall prepare the transformation detailed parameters for each VRF that shall be transformed. Such As:
     - IPv6 VRF naming conventions
     - IPv6 VRF Route Distinguisher
     - IPv6 VRF Route Target
     - Physical ports on which the IPv6 VRF shall be applied
     - Logical ports on which the IPv6 VRF shall be applied
     - IPv6 addressing for each site for each interface in that site associated with that VRF
     - IPv4 policies associated with the particular VRF that shall be transformed to IPv6 policies
     - IPv6 to IPv4 Transition needed (Yes or No) If yes determine the transition method (tunneling, dual stack, NAT)
7. Process Service Transformation
8. On that step the platform shall process the real calls configuring the network devices with the parameters specified above.
9. Process Service Transformation
10. Once the calls finish the system will complete the transformation and will update the network inventory database.

Only the senior ISP designers and OSS consultants have to be able design service transformation algorithms. The workflows most commonly will be executed by the NOC technicians through a simplified GUI interface. NOC operators shall have a GUI interface able to give them information about the start and finish times and dates, about the % of completeness and the relationship between the different tasks of the workflow. If we make a relationship with the Project Management we will find that such information is present in the common gantt charts. So the GUI interface will look like a GANTT chart.

5 Software Implementation

Currently the project is still in its specification stage. The project members responsible for that stage are Network Architects and are trying to specify the framework architecture features staying away from the software implementation part of the process. Once the specification is finalized the project will be transferred to the Software solution architects team.

However it is clear that the most appropriate software architecture for such solution is the Service Oriented Architecture. The Business Transformation Logic fits naturally to the SOA principles. The inventory and the service transformation might also be easily integrated as a web services next to the BTL. The tricky part of the software tools and technologies selection process is the exact software selection criteria [13] [14]. The software that will be used for the solution creation shall comply with the following criteria:

- Viability - Is the product widely used, and does it enjoy a strong user community? Is the solution well documented? Are sufficient development resources committed to the project?
- Architecture - Is the architecture of the product complementary to the other products we are evaluating? Is it well documented and logical, and does it adhere to common best practices and patterns?
- Monitoring and Management - Does the product provide off-the-shelf monitoring and management tools?
- Extensibility - Can the off-the-shelf solution be extended to add new functionality? Does a pluggable framework exist for adding new functionality?
- License - This is a sensitive topic, but we want to consider mostly products that are licensed using one of the common open source licenses: GPL, LGPL, BSD, Apache, or Mozilla Public License. We want to avoid, if possible, “free” or “community” versions that retain restrictions in usage or modification.

6 Conclusion

In this paper are presented the ideas behind the IP 4 to 6 transformation project. It aims to deliver a framework for IPv4 to IPv6 migration and to fulfill scientific gaps in the current OSS solutions. The framework will be released as an Open Source Product and consists of several major components: Business Transform Logic (BTL), Network Inventory and Service Transformation. Each of these components extends the state of the art products in its business field and delivers a scientific and practical value beyond current state of the art.

The project is in its specification phase. Despite the efforts of the project authors it may fail due to its complexity, the lack of resources and also due to the fact that the migration towards the new protocol might never happen. The project authors are trying to mitigate these reasons through:

Complexity - The best way to fight with it is to prepare a proper specification that might be easily followed by the rest of the project members.

The project has no financial sponsors and the project team members are participating in it in as volunteers in their spare time. From one side this is a problem but from the other it is advantage due to the fact that the project community are participating a number of professionals with different sets of knowledge instead of a fix number of employees. New Bulgarian University provides the base and the IT resources needed by the 4TO6TRANS team the project initial start.

Migration to IPv6 might be delayed or might never happen. Nevertheless the project aims to deliver an open source Fulfillment OSS solution that will bring benefits also to the existing IPv4 networks. It might be used not only for transforming IPv4 to IPv6 based services but also for transforming a current services and networks to something newer and better related with the future internet.

References

1. Ripe, IP Allocations, http://www.ripe.net/info/info-services/addressing.html
2. Postel, J. , ‘Internet Protocol’, RFC 791, IETF (1981)
3. Deering, S., Hinden, R., Internet Protocol, Version 6 (IPv6) Specification, RFC 2460, IETF (1998)
4. Deering, S., Hinden, R., Internet Protocol Version 6 (IPv6) Addressing Architecture, RFC 3413, IETF (2003)
5. Westerinen, A. et al., ‘Terminology for Policy-based Management’, RFC 3198, IETF (November 2001)
6. Moore, B. et al., ‘Policy Core Information Model – Version1 Specification’, RFC 3060, IETF (2001)
7. Perlman, R.,‘Inter-connections: Bridges and Routers’, Addison – Wesley (1992)
8. Comer, D., ‘Internet working with TCP/IP. Volume1. Principles, Protocols and Architecture’, Prentice-Hall (1995)
9. Stallings,W.,‘High-speed Networks, TCP/IP and ATM Design Principles’, Prentice-Hall (1998)
10. Jacquenet Chr., Bourdon G., Boucadair M., “Automation and Dynamic Provisioning Techniques in IP/MPLS Environments”, Wiley (2008)
11. GB921 Business Process Framework Release 8.0, TMFORUM (2009)
12. Clemm, Al., Network Management Fundamentals, Cisco Press (2006)
13. Hewitt, E., Java SOA Cookbook, OReill (2009)
14. Davis, J., Open Source SOA, Manning (2009)