www.euro6ix.net

Title:	Document Version:
Deliverable D4.4A **Report on the Second Year Network and Application Research Activities**	0.7

Project Number:	Project Acronym:	Project Title:
IST-2001-32161	Euro6IX	European IPv6 Internet Exchanges Backbone

Contractual Delivery Date:	Actual Delivery Date:	Deliverable Type* - Security**:
31/12/2003	04/02/2004	R – PU

* Type: P - Prototype, R - Report, D - Demonstrator, O - Other
** Security Class: PU- Public, PP – Restricted to other programme participants (including the Commission), RE – Restricted to a group defined by the consortium (including the Commission), CO – Confidential, only for members of the consortium (including the Commission)

Responsible and Editor/Author:	Organization:	Contributing WP:
David Fernández Antonio F. Gómez-Skarmeta	UPM UMU	WP4

Authors (organizations):

Diego Acosta (UPM), Eduardo Azañón (TID), Alain Baudot (FT), Tim Chown (UoS), Olivier Corre (6WIND), Fraser Christie (BT), Tomás de Miguel (UPM), Alvaro Díaz (UPM), Miguel A. Díaz (Consulintel), Xiang Fei (UoS), Juan L. Fernández (UPM), Juan A. Fernández (UPM), Aurora Ferrándiz (TID), Idir Fodil (6WIND), Felix J. García (UMU), Rüdiger Geib (TSN), Miguel Gómez (UPM), Olivier Goulpeau (6WIND), Ignacio Grande (TID), Jean-Mickael Guerin (6WIND), Miguel A. Morales (Consulintel), Vladimir Ksinant (6WIND), Gabriel López (UMU), Jesús López (TID), Rafael Marín (UMU), Cesar Martín (nGn), Rafael Martínez-Torres (nGn), Gregorio Martínez (UMU), Mario Morelli (TILAB), Jesús Muñoz (nGn), Karen E. Nielsen (TED), Cesar Olvera (Consulintel), Jordi Palet (Consulintel), Carlos Parada (PTIN), Cristina Peña (TID), María J. Perea (UPM), Juan Quemada (UPM), Carlos Ralli (TID), Alain Ritoux (6WIND), Tomas Robles (UPM), Carlos Rodrigues (PTIN), Francisco Romero (TID), José Luis Rubio (UPM), Antonio Tapiador (UPM), Ruth Vázquez (TID), Alvaro Vives (Consulintel).

Abstract:

This document summarizes the results of the research and innovative tasks carried out during the second year of Euro6IX project in the context of WP4 activities A4.1 (Advanced Network Services Design and Evaluation) and A4.2 (IPv6 Application Development). WP4 main aim is the design, consolidation and validation of an IPv6 based integrated end user service scenario, which integrates advanced network services and applications into a common framework. During year 2, WP4 work has concentrated on the integration of the most important network and application activities, which were consolidated during the first year.

Keywords:

AAA, AGWS, Code Porting, DHCPv6, DIAMETER, DiffServ, DNSSEC, ECN, IDS, Instant Messaging, IntServ, IPsec, ISABEL, IX, Jabber, Looking Glass, M6Bone, Magalia, Mobile IPv6, Mobility, MRTG, Multicast, Multihoming, NAT-PT, PANA, Peer-to-Peer, PERL, PINE, PKIv6, Porting guide, P2P, QoS, RADIUS, Roaming, Route Server, RPSLng, Security, Security policy, SIG, SIP, Source address selection, TCL, Topaz, TSIG, Tunnel Broker, VNC, VNUML, VPN, WRED, Xges.

© Euro6IX Consortium

Revision History

The following table describes the main changes done in the document since created.

Revision	Date	Description	Author (Organization)
v0.1	28/10/2003	Document creation	David Fernández (UPM) Antonio F. Gómez-Skarmeta (UMU)
v0.2	30/11/2003	Integration of sub-activity summaries	David Fernández (UPM) Antonio F. Gómez-Skarmeta (UMU)
v0.3	18/12/2003	Final corrections to sub-activity summaries	David Fernández (UPM)
v0.4	22/12/2003	Introduction and conclusions added	Juan Quemada (UPM) David Fernández (UPM)
v0.5	05/01/2004	Minor corrections	David Fernández (UPM)
v0.6	30/01/2004	Final review	Jordi Palet (Consulintel)
v0.7	04/02/2004	Minor changes to improve some pictures quality	Jordi Palet (Consulintel)

Executive Summary

This document summarizes the results of the research and innovative tasks carried out along the second year of Euro6IX project in the context of WP4 activities A4.1 (Advanced Network Services Design and Evaluation) and A4.2 (IPv6 Application Development).

WP4 main aim is the design, consolidation and validation of an IPv6 based integrated end user service scenario, which integrates advanced network services and applications into a common framework, which we call the "Euro6IX Integrated Service Framework". Therefore WP4 considers the integration of several advanced network services together with applications, which exploit the new features as one important goal.

During the second year of the project, WP4 work has concentrated on the integration of the most important network and application activities that were consolidated during the first year. Therefore the results presented in this deliverable include the first products of the integration effort which shall lead to the "Euro6IX Integrated Service Framework". As the project advances those results shall be deployed and validated in the Euro6IX test-bed.

The deployment on the Euro6IX test-bed of new architectures and features made possible by IPv6 is an important priority for WP4. Therefore tools for management and operation of the test-bed, as well as experimentation with new features such as autoconfiguration, provider independent addressing, multihoming, route servers, etc, are very thoroughly addressed.

The introduction of security into the future Internet is considered one of the main challenges and, therefore, Euro6IX includes activities dealing with security from several perspectives and with different approaches. Many sub-activities such as IPsec-VPNs, PKIv6, DNSSEC or AAA address network security features, which have to be integrated in parallel into application frameworks, such as SIP, the ISABEL CSCW application, unified instant messaging, as well as groupware applications.

As a result of the evaluation, which took place in year 1, mobility, multicast and QoS components were considered mature enough to be transferred to the Euro6IX test-bed. Therefore in year 2 the work has focused on the definition of integration scenarios that allow the deployment of such services over the Euro6IX test-bed. Activities on IPv4/v6 interworking and multihoming were started in year 2, due to the importance of those topics and the interest rose.

The work on porting, enhancement and tuning of applications has continued with the aim of supporting the test-bed deployment, validating the network services and supporting the trials. Those activities are complemented with the porting of generic software libraries and packages, as well as applications, which are considered important for the transition to IPv6.

The document is organized as follows. Section 1 introduces WP4 general context, objectives and organization, as well as summarizes the general objectives of A4.1 and A4.2 activities for Y2. Section 2 and 3 detail the objectives and results of the different sub-activities under A4.1 and A4.2 respectively. Finally, Section 4 shows some conclusions about the work done during the second year.

Table of Contents

1. Introduction ... 8

1.1 Structure of WP4 ... 9

1.2 Structure of this Document ... 10

1.3 Summary of A4.1: Advanced Network Services Design and Evaluation ... 10

1.4 Summary of A4.2: IPv6 Application Development ... 11

2. Results on Advanced Network Services ... 14

2.1 Security Services (A4.1-1) ... 14
2.1.1 Deployment of DNSSEC and PKI ... 14
2.1.2 End-to-End Security: VPN and SSL ... 19
2.1.3 Security Policy ... 22
2.1.4 IPv6 Filters and Firewall Functions ... 27
2.1.5 Security of IPv6 Routing Protocols ... 29
2.1.6 Network Security ... 31
2.1.7 Intrusion Detection Systems ... 36
2.1.8 Transition Mechanisms and Security ... 41

2.2 Mobility and Roaming Evaluation for MIPv6 (A4.1-2) ... 45
2.2.1 AAA for IPv6 Networks ... 45
2.2.2 UMU Test-bed Based on PDAs and MIPv6 ... 51
2.2.3 Secured MIPv6 ... 53
2.2.4 RADIUS and PPPv6 Integration Test-bed for RAS Service on Linux ... 54
2.2.5 TID Test-bed Based on PDAs and MIPv6 ... 57
2.2.6 NPF Framework and Service API Standard for Mobile IPv6 ... 60

2.3 QoS Deployment within IX (A4.1-3) ... 64

2.4 Internetworking v4/v6 (A4.1-4) ... 69
2.4.1 Tunnel Broker ... 69
2.4.2 Forwarding Protocol 41 in NAT Boxes ... 79
2.4.3 NAT-PT ... 81

2.5 WP2/WP4 Common Evaluation Activity (A4.1-5) ... 86
2.5.1 Multicast ... 86
2.5.2 Multihoming ... 89

3. Results on Application Development ... 93

3.1 Advanced Applications (A4.2-1) ... 93
3.1.1 Multimedia Service Provision Platform using SIP ... 93
3.1.2 IPv6-Capable ISABEL ... 96
3.1.3 IPv6 Instant and Unified Messaging ... 101
3.1.4 Groupware Applications ... 105
3.1.5 IPv6 Sharing Information Tool ... 108

3.2 Network Management and Operation Tools (A4.2-2) ... 112
3.2.1 Magalia ... 112
3.2.2 Topaz ... 116

3.3 IX Support Tools (A4.2-3) ... 119
3.3.1 Route Server ... 119

3.3.2 Looking Glass 123

3.4 Code Porting (A4.2-4) 125

3.4.1 Application Migration 125

3.4.2 Code Porting Guide 128

4. Summary and Conclusions 130

5. References 132

Table of Figures

Figure 1-1: Relations between WP4 Activities 10
Figure 2-1: TSIG Model 16
Figure 2-2: Design of PKI CERT-Publisher 17
Figure 2-3: Scenario 1 – Root CA and Name Server are Together in the IX 18
Figure 2-4: Scenario 2 – Root CA is Out 18
Figure 2-5: UMU Policy-Based Network Management 23
Figure 2-6: The Policy Management Process in the UMU-PBNM Architecture 24
Figure 2-7: The Policy Monitoring Process in the UMU-PBNM Architecture 26
Figure 2-8: IPv6 Filters and Firewall Functions (I) 27
Figure 2-9: IPv6 Filters and Firewall Functions (II) 28
Figure 2-10: RIPng or OSPFv3 + IPsec Scenario 30
Figure 2-11: Traffic Control in the Carrier's Network 32
Figure 2-12: Traffic Control in the ISP's Network 33
Figure 2-13: Traffic Control in the IPv6 Internet Exchange 33
Figure 2-14: IDS Scenario 39
Figure 2-15: SNORT Summary Log 39
Figure 2-16: IPsec Solution 44
Figure 2-17: Unique Domain Radius and IPv6 UMU Test-bed 46
Figure 2-18: Multi-Domain Radius and IPv6 UMU Test-bed 46
Figure 2-19: Prefix Delegation & Radius v6 Scenario 47
Figure 2-20: Hot Spot & Radiusv6 Scenario 48
Figure 2-21: Translator RADIUS-DIAMETER Test-bed 48
Figure 2-22: Possible Experiment Mobile IPv6 Access Control 50
Figure 2-23: UMU Test-bed 51
Figure 2-24: MIPv6 Scenario 53
Figure 2-25: Implemented RAS Test-bed 55
Figure 2-26: TID Test-bed 57
Figure 2-27: IPv6 Mobility Options 58
Figure 2-28: NPF and IETF Standards Relation 61
Figure 2-29: The NPF Software Architecture Framework 62
Figure 2-30: Euro6IX QoS Architecture 65
Figure 2-31: IntServ over DiffServ Model 66
Figure 2-32: QoS on Demand Prototype 67
Figure 2-33: Tunnel Broker Architecture 70
Figure 2-34: TB Behind NAT Configuration 72
Figure 2-35: IPv6 over PPP over UDP Approach 73
Figure 2-36: UTMP Entities 74
Figure 2-37: UTMP Interactions 74
Figure 2-38: First UMU-TB Design 76
Figure 2-39: Tunnel Broker Structure 77
Figure 2-40: Tunnel Request Page 77
Figure 2-41: Basic Proto-41 Forwarding Scenario 79
Figure 2-42: NAT-PT – IXv4 81
Figure 2-43: NAT-PT – IXv6 82
Figure 2-44: NAT-PT – IXv4 – IXv6 83
Figure 2-45: NAT-PT – IX Dual Stack 83
Figure 2-46: NAT-PT Network 84
Figure 2-47: Transition Scenario for Euro6IX Connection to M6Bone Network 87
Figure 2-48: UPM's Multihoming Web Demonstration Scenario 91
Figure 3-1: ISABeLE-SIP Gateway Architecture 95
Figure 3-2: Connectivity Requirements after Y2 97
Figure 3-3: Sample Network Topology 97
Figure 3-4: ISABEL over IPsec Tunnels 98
Figure 3-5: Session Management Process 99
Figure 3-6: Work Done for IPv6-Capable Sub-activity During Y2 100
Figure 3-7: Hybrid Model 102
Figure 3-7: IPv6 Instant and Unified Messaging Integration 104
Figure 3-9: ISABEL Web Antenna Architecture 106

Figure 3-10: General Image of the GUI 109
Figure 3-11: General Image with Search Option Selected 109
Figure 3-12: Connections between Client Daemon and GUI 110
Figure 3-13: Xges with Edit Support 113
Figure 3-14: Message Paths Propose for Euro6IX 113
Figure 3-15: Travel of an Event and Forwarding of this Event 114
Figure 3-16: New Architecture of Topaz 117
Figure 3-17: Rules Manager 117
Figure 3-18: List Manager 117
Figure 3-19: Communication between Topaz and Magalia System 118
Figure 3-20: IX Model 121
Figure 3-21: BGP peering scheme 122
Figure 3-22: Looking Glass Interface 124
Figure 3-23: Perl and IPv6 Integration 125
Figure 3-24: Network Layer Modified by nGn on PINE Context 126
Figure 3-25: Structure of TCL Interpreter and Network Component Involved 126
Figure 3-26: Structure of MRTG (green modules modified initially by nGn) 127

1. INTRODUCTION

This Deliverable describes the results obtained after the first 2 years of Euro6IX in the network and application research activities of Work Package 4 "Associated Research Activities, Trials and Evaluation".

The Euro6IX test-bed has been conceived as a small-scale model of the Internet, which includes the main components of the actual global Internet such as a backbone of IX nodes (traffic exchange nodes) operated by different telcos and interconnecting several AS's (Autonomous Systems) which represent regional networks containing a variety of providers and access technologies.

WP4 has as its main aim, the design, consolidation and validation of an integrated end user service scenario, which integrates advanced network services and applications into a common framework, which will be called the "Euro6IX Integrated Service Framework". Therefore WP4 considers the integration of several advanced network services together with applications, which exploit the new features as one important goal.

During Y2 WP4 has concentrated on further developing immature components, as well as on the integration of the most important network and application activities which were consolidated during the first year, where important components needed for the deployment of the Euro6IX test-bed, as well as for the "Euro6IX Integrated Service Framework" were consolidated. Therefore the results presented in this deliverable include the first integration results which shall lead to the "Euro6IX Integrated Service Framework". As the project advances those results shall be deployed and validated in the Euro6IX test-bed.

The deployment on the Euro6IX test-bed of new architectures and features made possible by IPv6 is an important priority for WP4. Therefore tools for management and operation of the test-bed, as well as experimentation with new features such as autoconfiguration, provider independent addressing, multihoming, route servers, etc, are very thoroughly addressed.

The introduction of security into the future Internet is considered one of the main challenges and, therefore, Euro6IX includes activities dealing with security from several perspectives and with different approaches. All this activities must provide the foundations for assembling and validating at the end of the project an integral security approach based on IPv6. The introduction of security implies advanced network security feature deployment, as well as adaptation and tuning of applications. Therefore many sub-activities such as IPsec-VPNs, PKIv6, DNSSEC or AAA address network security features, which have to be integrated in parallel into application frameworks, such as SIP, the ISABEL CSCW application, unified instant messaging, as well as groupware applications.

As a result of the evaluation, which took place in year 1, mobility, multicast and QoS components were considered mature enough to be transferred to the Euro6IX test-bed. Therefore in Y2 the work has focused on the definition of integration scenarios that allow the deployment of such services over the Euro6IX test-bed. Activities on IPv4/v6 interworking and multihoming were started in year 2, due to the importance of those topics and the interest rose.

The work on porting, enhancement and tuning of applications has continued with the aim of supporting the test-bed deployment, validating the network services and supporting the trials. Those activities are complemented with the porting of generic software libraries and packages, as well as applications, which are considered important for the transition to IPv6.

1.1 Structure of WP4

To achieve the goals assigned to WP4 in Euro6IX, the work package has been divided in four activities which have to work in close cooperation and which are:

- **A4.1 – Advanced network services design and evaluation:** Groups all the research, development, integration and validation of advanced IPv6 network services which are candidates for integration in the Euro6IX test-bed, such as mobility, multicast, anycast, security, multihoming, renumbering, QoS/CoS, etc.

  Rather than considering the services in isolation, this activity will focus on the integration of all those services into the common operational framework of the Euro6IX test-bed, which will be called the "Euro6IX Integrated Service Framework". The integration will be performed considering:
  - Incompatibilities and interaction problems of the integration of services.
  - Scalability of services across the IX core and impact of the service in the IX architecture.
  - Relevance for the deployment of IPv6.
- **A4.2 – IPv6 application development:** Groups all the research, development, adaptation, enhancement, maintenance and validation of applications, which are considered most relevant for the successful deployment of the Euro6IX test-bed, the realisation of the Euro6IX trials and for facilitating the transition to IPv6.

  This activity identifies the following important goals:
  - Application porting, adaptation, development or tuning as needed for the deployment, operation and management of the Euro6IX test-bed.
  - Application porting, adaptation, development or tuning as needed for the validation and assessment of the "Euro6IX Integrated Service Framework" development done in Activity A4.1.
  - Application porting, adaptation, development or tuning as needed for the realisation of the Euro6IX public and internal trials.
  - Application development when considered important to facilitate the transition to IPv6 or to extract the potential of IPv6 advanced network services.

  This activity will mainly perform the porting and adaptation of existing applications to meet the above-mentioned goals.

  Additionally, new applications will be developed when no existing one is considered able to exploit the features offered by IPv6.

  Finally the activity will also consolidate all the experience gained in making the transition from IPv4 to IPv6 in the form of a Guide for the porting and transition of applications to IPv6, which is a continuation of the work started in the LONG IST project.
- **A4.3 – Trials and evaluation:** Groups all the definition, planning, realization and evaluation of trials to be performed over the Euro6IX test-bed. Trials will include internal trials to evaluate and assess user services in an end-to-end user scenario, as well as public trials, which will demonstrate the new capabilities of IPv6.

- **A4.4 – Legal aspects:** Deals with the impact of IPv6 in Privacy/Civil Liberty, data protection, and IPR issues.

It is important to remark that activities A4.1 and A4.2 will provide the network service, application development and integration as needed for the realization of the trials in A4.3. Therefore all three activities will work in close cooperation.

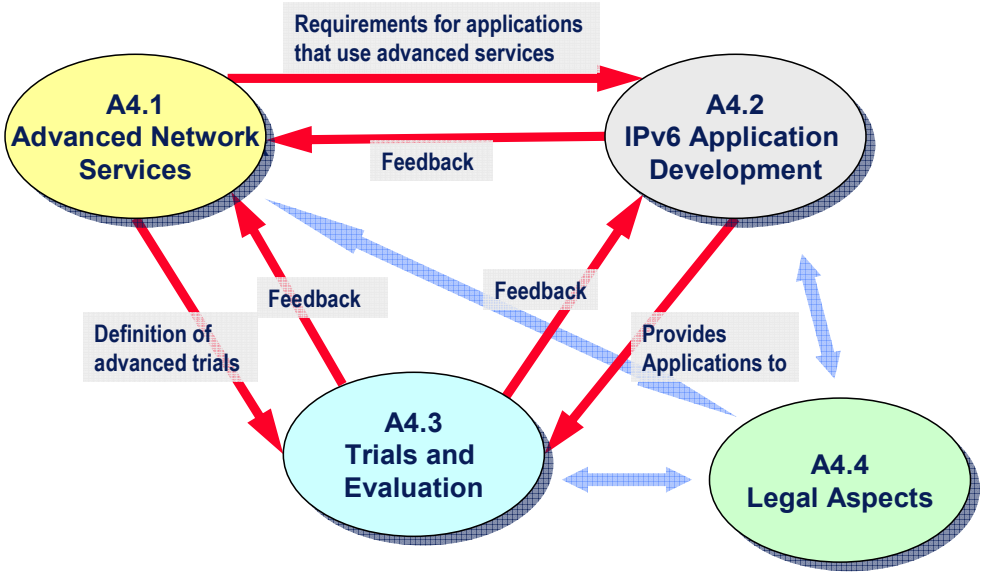

Figure 1-1: Relations between WP4 Activities

Figure 1-1 shows graphically the relations between WP4 activities.

1.2 Structure of this Document

This document summarizes the results obtained during Y2 in A4.1 Advanced network services design and evaluation and A4.2 IPv6 application development. The document has been structured therefore following the work and management structure of those activities. The results of each activity are described in each of the two next chapters and each chapter contains one section for each of the sub-activities. This draft deliverable contains only an overview of the work done. It will be complemented at the end of Y2 with the associated "Technical Reports" which further describe the details about all the work carried out under each sub-activity.

1.3 Summary of A4.1: Advanced Network Services Design and Evaluation

A4.1 groups all the research, development, integration and validation of advanced IPv6 network services and features, which are candidates for inclusion into the Euro6IX Integrated Service Framework. Those services include mobility, various security components, multicast, anycast, multihoming, renumbering, QoS/CoS, etc. The summary of results produced in A4.1 during Y2 is as follows:

- **Security framework:** The first project year did focus on isolated experiments and developments in: static IPsec/IKE VPN deployment, firewalls, set up of a Java based PKIv6/v4 service, feasibility analysis of DNSSEC, and AAAv6. During the second year Euro6IX is focussing on the introduction of security in the Euro6IX test-bed by: integrating PKI services into IXs, linking PKI and IPsec, further working on Firewalls which are considered still immature, secure routing protocols, analysis of deployment scenarios of DNSSEC and in producing network security guidelines.
- **Mobility:** The first project year did focus on isolated experiments and developments in micro and macro mobility with IPv6. As the implementations where considered mature enough to start deployment over the Euro6IX test-bed, during the second year this WP is focussing on the deployment of a multidomain mobility scenario attached to the IXs, including Mobile IPv6, authentication based on AAA, Radius, Diameter, etc.
- **IPv4/v6 Interworking:** During the Y2 work has been done on the evaluation of IX attached IPv4/v6 interworking functions. Therefore the most suitable transition mechanisms have been selected for deployment in the IXs. NAT-PT has been deployed in BT´s IX. Finally, several Tunnel Broker scenarios have been analysed using IPv6 over UDP. A draft of protocol 41 forwarding mechanism of IPv6 over IPv4 tunnels over NAT has been contributed to IETF.
- **Multihoming:** During Y2 WP4 has worked on the introduction of multihoming capabilities in the USAGI Linux Kernel, which has only a partial implementation of RFC3484. The Linux Kernel has been enriched with a PCT (Policy Configuration Table), which allows the source host multihoming where each destination is contacted with the right source address. Work is ongoing on the integration of router advertisement, which will continue on year 3.
- **Multicast and Anycast:** During the first year WP4 did test multicast scenarios over IPv6 with successful results. Anycast is considered still immature. During the second year WP4 has focussed on the introduction of a multi IX multicast scenario in the Euro6IX test-bed. The activity has been transferred during Y2 to WP3.
- **QoS over IPv6:** During the first year WP4 has deployed and tested QoS implementations for DiffServ, which seem stable and mature. During the second year WP4 has focussed on the development of a model, which includes SLAs for premium services, and on the introduction of this model in a multi IX DiffServ scenario for the Euro6IX test-bed. The activity has been transferred during Y2 to WP3.

1.4 Summary of A4.2: IPv6 Application Development

A4.2 groups all the research, development, adaptation, enhancement, maintenance and validation of applications which are considered of relevance for the successful deployment and validation of the Euro6IX test-bed and Integrated Service Framework, as well as for facilitating the transition to IPv6. The summary of results produced in A4.2 during Y2 is as follows:

- **Advanced Applications:** The sub-activities described below have been carried out in order to support the creation of a realistic set of end user services, used to validate the Euro6IX Integrated Service Framework, as well as for the realization of trials.
    - *Multimedia services provision platform using SIP*: During Y2 A4.2 has focussed on the development of an ISABEL-SIP gateway that allows interworking with ISABEL sessions from SIP videoconferencing clients. The ISABEL-SIP gateway has been and demonstrated in the Madrid 2003 Global IPv6 Summit. The use of SIPv6 based ISABEL event organisation has been analysed.

- *IPv6 capable ISABEL*: During Y2 A4.2 has continued the work on integrating ISABEL into the Euro6IX service framework in order to make use of features, such as security, QoS or mobility. ISABEL has been integrated with the Euro6IX security framework. The integration has been made by developing XEDL 1.8 (a new version of EDL, the "sEssion Definition Language" of ISABEL), which has been defined in XML using XML Schema and which includes generic service components, such as IPsec associations. XEDL 1.8 has been already defined and integrated into the Euro6IX service framework that allows defining and launching secure ISABEL platforms. This version has been validated in several experiments.
- *IPv6 Unified Instant Messaging Systems*: During Y2 A4.2 has continued the work on investigating the benefits of using IPv6 in unified and instant messaging, focussing mainly on Jabber. After the porting to IPv6 done in year 1, Jabber has been used to deploy IM scenarios, which have been used to debug the porting and to generate a first P2P prototype.
- *Groupware applications*: During Y2 A4.2 has continued the work of integrating the AGWS groupware tools, with ISABEL and with the security framework, which is finished now. The VNC remote application sharing tool, ported during year 1 has been used to construct an ISABEL Web Antenna, which allows high quality streaming of ISABEL sessions over standard Web browsers.
- *IPv6 Sharing Information Tool*: During Y2 A4.2 has started work on P2P file sharing tool, which should exploit the benefits of the extended address space of IPv6. A first prototype of the tool that uses multicast has been finished.

- **IPv6 Network Management and Operation Tools:** During Y2 A4.2 has continued the work on network management and operation tools, which support the deployment of the Euro6IX test-bed. The first tool is an SNMP network management tool called Magalia, which implements a distributed management model adapted to the Euro6IX IX architecture. During the second year Magalia has been enhanced with shared management with visibility control. The second one is TOPAZ a network intrusion detection system that implements IPv6 specific detection patterns. The first working prototype of TOPAZ has been finished in Y2. The tools have been transferred to WP3 to allow management and analysis of the Euro6IX test-bed.
- **IX Support Tools:** During Y2 A4.2 During Y2 Euro6IX has continued the work on the creation of a simulation environment of IX functions, the implementation of the route server function over an open source routing daemon, as well as in the development of a looking glass function to be deployed associated to the IXs.

  The simulation environment developed during Y1, for analyzing complex IX configurations, has been greatly improved during Y2. This tool, based on User Mode Linux (UML) virtualization software and enriched with an XML language to define simulation scenarios, has proven very useful to facilitate the analysis and configuration of the various IX models considered in WP2. The simulation environment has been used to investigate IX scenarios based on route servers and IX based address assignment.

  Apart from that, work has been invested on modifying and completing the route server implementation of the open source routing daemon selected (quagga), as several important deficiencies were found.

  A looking glass application to be deployed as an IX function has been consolidated and transferred to WP2/WP3. It allows Web access to several network analysis tools to be attached to each IX, such as network trace tools, DNS analysis tools, BGP analysis tools and routing display commands.

Finally, although the introduction of security in routing protocols working on IXs has been initiated, the main work will be developed during Y3.

- **Code Porting:** During Y2 A4.2 has continued the work on porting libraries or software packages. In particular the porting of PERL, MRTG, PINE and TCL has been undertaken. The porting have been contributed to the projects for inclusion on the standard software releases and are made available also by means of the Euro6IX repository. In addition, the IPv6 Porting Guide produced in the LONG project has been transferred to Euro6IX after the end of LONG, where it is being maintained. An improved version of the IPv6 Porting Guide has been generated.

2. RESULTS ON ADVANCED NETWORK SERVICES

This section summarizes the main results obtained in WP4 in the context of activity A4.1, Advanced Network Services, during the second year of Euro6IX project. The subsections below just outline the results of the different sub-activities under A4.1. More detailed information can be found in the Technical Reports associated to each sub-activity (see Section 5).

2.1 Security Services (A4.1-1)

2.1.1 Deployment of DNSSEC and PKI

Partners

UMU, UPM, Consulintel

Objectives

- Set up a DNSSEC pilot experiment between several partners in order to gain experience in the management procedures and generate recommendations to extend the service to the whole Euro6IX network.
- Design and develop the PKIv6 module to support the publication of certificates by DNSSEC.
- Define and evaluate an IX environment to publish digital certificates by the updating DNS data dynamically.
- Study DNSSEC functionalities within the IX for secure publication of certificates.

Description

In view of the importance of the deployment of secure DNS services based on DNSSEC and other extensions proposed by IETF, it was decided to start a new sub-activity inside WP4 during the second semester of the first year to study the initiatives, standards and implementations available, as well as to set-up a preliminary test-bed, with the long-term objective of deploying a secure DNS service over the Euro6IX network.

Besides, work in this sub-activity has concentrated on studying the means to publish certificates using DNSSEC and PKIv6, in order to allow users and applications to retrieve a certificate with a simple DNS request based on FQDN (Fully Qualified Domain Name) of the certificate. When an application (or user) wants to retrieve a certificate linked to a known FQDN, the application can send a DNS query of type CERT and the DNS resolver will return the certificate.

Both DNSSEC and PKIv6 are value-added services that complete the variety of new possibilities of the Layer 3 Exchanges. So it is consistent the incorporation of the use of DNSSEC in PKIv6 to manage (publish/revoke/renew) certificates.

Relevance

PKIs are needed in the IPv6 world to offer cryptographic features to security services like HTTPS, IPsec, etc. On the other side, DNSSEC is used to secure DNS transactions between clients and servers and among servers, as well as to store cryptographic information about the entities stored in it. Therefore, DNSSEC can be naturally used by a PKI to store the security information of the PKI clients, offering a worldwide distributed cryptographic storage mechanism. Using DNSSEC to store Public Key Certificates (PKC) and Certificate Revocation Lists (CRL), an IPv6 user can easily find the public certificate associated to other person/entity by means of a simple DNS request, in order to exchange information in a secure way.

Results

During the second year, several important results have been achieved:

- A limited DNSSEC pilot experiment has been set-up between three partners (UPM, UMU and Consulintel). Experience gained with the pilot has led to a proposal to set-up a Euro6IX wide pilot to be carried out during the rest of the project in coordination with WP3. The partners involved had gained a very useful know-how, and as a result the idea of a help tool appeared. This tool will help both to manage the signed keys and the process of joining the DNSSEC pilot to new partners. The development of this tool has already started and it will be finished at the beginning of Y3.
- A complete set of DNSSEC examples has been developed using the simple DNS emulation environment developed during Y1. These examples are based on a complete DNS hierarchy, including a root server for a fictitious TLD (.e6), three main domains (upm.e6, umu.e6 and consulintel.e6) and two delegated subdomains (dit.upm.e6 and dif.umu.e6). The examples have been selected to help to understand how DNSSEC and TSIG extensions work using IPv4 and IPv6. They have been developed using “bind” (version 9.3.0 is needed, as previous versions have problems when using IPv6 sub-interfaces) and have been documented including all name servers configurations, log traces and sniffer captures. They will be a valuable testing and training tool for extending the DNSSEC pilot experiment.

Related to the study about using DNSSEC to publish certificates, two models have been defined: *TSIG Model* based on shared secrets and *SIG Model* using public and private keys.

TSIG Model

The TSIG model is based on TSIG keys described in RFC 2845. This model has the main following elements:

- PKI CERT-publisher: It is a PKI module able to manage X.509 public key certificates (PKC) and CRL with one or several DNSSEC systems. It exchanges TSIG messages with the appropriate DNSSEC server.
- PKI TSIG Database: It is a PKI database containing a entry for each domain name in which there is a DNSSEC server allowing TSIG transactions. The basic structure of each entry will be the FQDN of every domain name allowing DNSSEC communications, a KEY used for authentication against DNSSEC server, the algorithm (ALG) used by this KEY and the KEY name. PKI CERT-publisher will access PKI TSIG Database.

Publishing certificates model is a way to publish, update and remove PKCs and CRLs in a DNSSEC server by a PKI. It allows final users and systems to obtain PKCs and CRLs easily to establish secure relationships. The way this model works is shown in Figure 2-1.

When a PKI wants to publish a user's certificate in a DNSSEC server it has to obtain the name of the domain that should store the PKC. The PKI CERT-publisher module is in charge of doing that, possibly by means of an Alternative Subject Extensions with a DNS attribute.

Once the PKI CERT-publisher knows the domain's FQDN, it consults the PKI TSIG Database to know if the user's certificate can be published in this domain. The PKI TSIG Database must contain an entry for this domain with the right KEY, ALG and KEY name and return them to PKI CERT-publisher.

Then, the PKI CERT-publisher does a NS request to its local resolver to know the IP address of the DNSSEC server that should publish the PKC. The local resolver obtains the AAAA record corresponding to the DNSSEC server and returns it to the PKI CERT-publisher. In order to do this, the local resolver should verify the chain of SIGs and KEYs objects received.

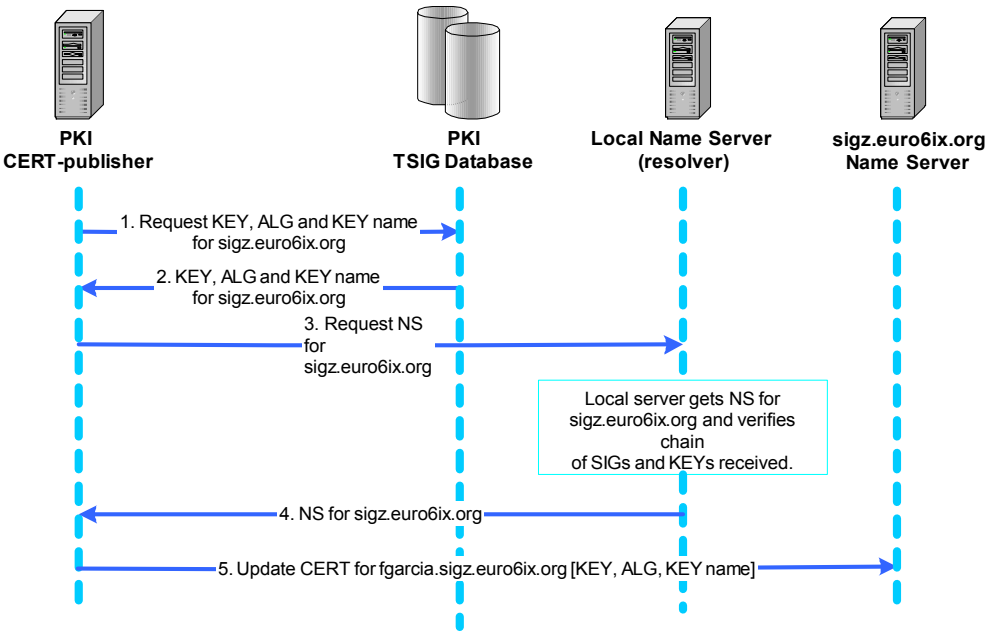

Figure 2-1: TSIG Model

When the PKI CERT-publisher knows where it should publish the certificate, it sends a Zone UPDATE TSIG message with the user's PKC using the KEY and ALG related to the server.

If the PKI CERT-publisher wants to make another certificate operation like remove or update a PKC or add or remove a CRL, the operational way is the same mentioned above with the appropriate TSIG operation.

SIG Model

The SIG model is based on SIG(0) keys described in RFC 2535 and RFC 2931. This model has the following main elements:

- PKI CERT-publisher: This entity is similar to the one previously explained, but now it does not maintain a shared secrets database; instead, it knows the public key of every DNSSEC server wants to communicate to.
- Local Name Server: The same as above.

In this model, the CERT-publisher tries to get its public key from the nameserver, in order to know if the publisher is able to communicate with the DNSSEC server. If the public key is stored it sends a SIG message to the DNSSEC server with the required operation.

UMU-PKIv6 supports TSIG Model

UMU-PKIv6 is able to interact with a DNS or DNSSEC server, which can store CERT Resource Record (RR) by means of the TSIG protocol. TSIG protocol allows a process to update the DNS information using shared symmetric keys. The other option based on SIG protocol, which allows the use of asymmetric keys, could be used, but nowadays there is not a stable implementation of this protocol. The TSIG model is included in a stable way in BIND 9.2 DNS server implementation. That is the main reasons why we have decided to use it.

TSIG model forces to UMU-PKIv6 to take a control of the symmetric keys shared with all the DNS/DNSSEC servers UMU-PKIv6 can interact with. For every allowed DNS/DNSSEC server, UMU-PKIv6 must know the zone name, the shared key name, the key algorithm and the key value.

Design of PKI CERT-publisher

General features:

- Every zone UMU-PKIv6 wants to have access to should be added to an internal database.
- The name server where the user's certificate will be published can be a default one or specified by the user.
- The user should include a SubjectAltName extension, with DNSname property, in the certificate with the user and domain name to publish the certificate.
- The user's certificate is published automatically when it is generated.
- The user can retrieve his certificate from the PKI web pages.
- From the CA application, the PKI administrators can enable/disable the CRLs publication.
- From the CA application, the PKI administrator can publish and remove the user's certificates.
- When a certificate is revoked it is not removed in the name server.

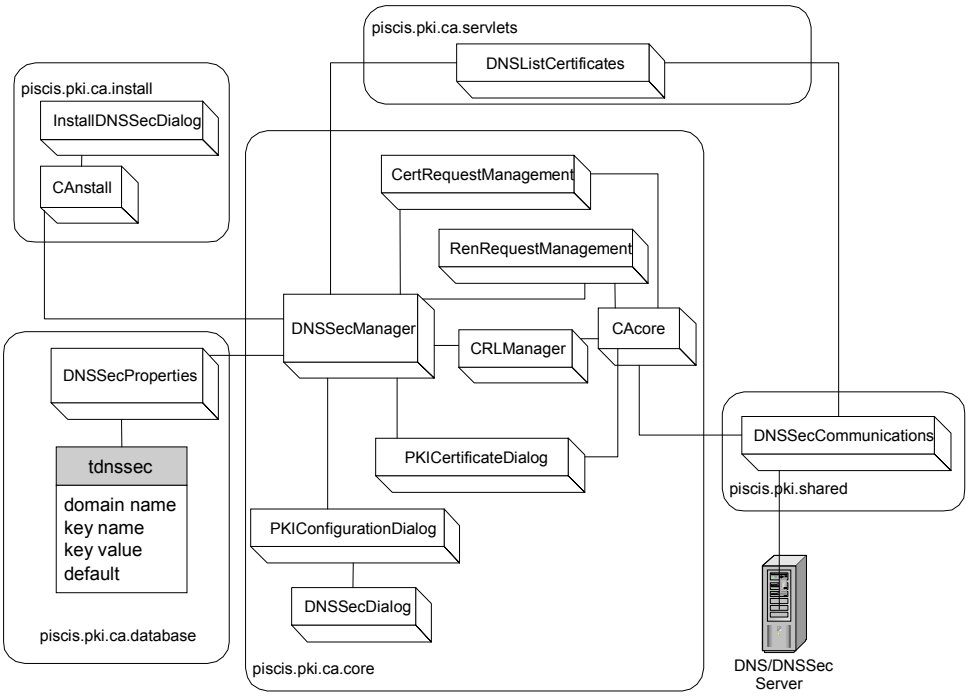

Figure 2-2: Design of PKI CERT-Publisher

The diagram in Figure 2-2 shows the java classes involved in the design. These classes are grouped by java packages.

IX Scenarios

Scenario 1: *Root CA and Name Server inside the IX*. In this case (Figure 2-3), the IX provides all L3 services. DNSSEC and PKI are IX services directly offered to IX customers.

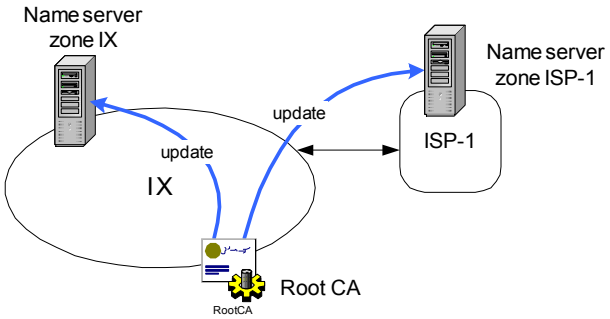

Figure 2-3: Scenario 1 – Root CA and Name Server are Together in the IX

ISPs can use the PKI and DNSSEC services offered by the IX, as well as allowing that the Root CA updates its DNS zone, publishing ISP certificates in its Name Server. The IX entities that implement this scenario are completely autonomous. Other IX entities could use its PKI and DNSSEC services and, therefore, depend on it.

Scenario 2: *Root CA outside the IX*. In this case (Figure 2-4), the IX entity does not provide the PKI service, but it still offer certificates through DNSSEC. The Root CA manages the certificates of all ISPs. Only previously authorized Root CAs can update the IX name servers.

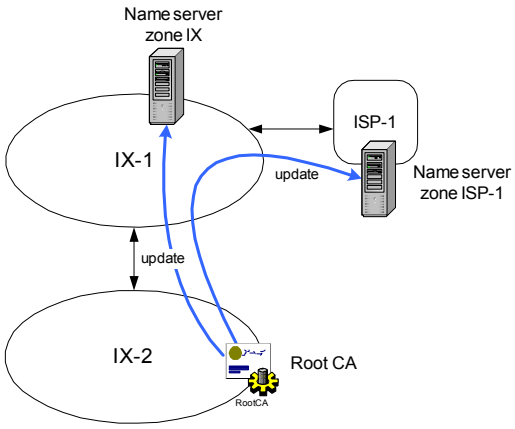

Figure 2-4: Scenario 2 – Root CA is Out

The IX entities that implemented this scenario are not autonomous. They depend on other IX entities providing the PKI service.

Future Work

- Deployment and evaluation of a stable IX environment.
- Extension of the DNSSEC pilot experiment to the whole Euro6IX network.
- Related to the above point, the development of a web interface-based tool to help in the key management procedures, in order to facilitate other partner's participation.

2.1.2 End-to-End Security: VPN and SSL

Partners

UMU

Objectives

- Study the End-To-End solutions with IKE/IPsec-based VPNs using certificates.
- Study the End-To-End solutions with SSL using certificates.
- Design solutions for project services with end-to-end security requirements.

Description

The architecture of IPv6 has been designed to provide globally unique addresses and so it contributes hard to end-to-end connectivity. End-to-end applications can transmit sensitive information over IPv6 and they must not worry about whom might see it. The major end-to-end secure technologies are IPsec-based VPN and SSL Protocol. A higher-level protocol can layer on top of the SSL Protocol or an IPsec-based VPN transparently, both ones are application protocol independent.

VPN based on the use of IPsec is the case of providing end-to-end security in the network layer between two hosts. IPsec is a mandatory component of IPv6 and so this technology has high relevance. IKE permits automatic key management in IPsec using public keys that may be offered by a Public Key Infrastructure (PKI). An important task objective is the study and evaluation of IPsec/IKE-based end-to-end VPN using PKIv6.

SSL (Secure Socket Layer) Protocol provides end-to-end security in an upper layer. SSL is an essential component of an end-to-end Web security solution. SSL Protocol allows the web server and web client (usually a navigator) to authenticate each other and to negotiate an encryption algorithm and cryptographic keys before the application protocol transmits or receives its first byte of data. The authentication is based on digitally signature and certificates. Another important task objective is the study and evaluation of end-to-end security using SSL Protocol and PKIv6.

Finally, we are going to evaluate the project services and applications with end-to-end security requirements, and design solutions according to them.

Relevance

There are IPv6 services with end-to-end security requirements so it is necessary to study the technologies that provide end-to-end security. IPsec is mandatory component of IPv6 so IPsec/IKE-based end-to-end VPN is the major technology together with SSL protocol.

Results

<u>End-To-End solutions with IKE/IPsec-based VPNs using certificates</u>

The main implementations supporting IKE/IPsec using certificates are KAME and FreeS/WAN. Both implementations are open source solutions.

- KAME Project is a prominent open-source effort to provide a free IPv6 and IPsec stack for BSD variants.

- FreeS/WAN Project offers an open source implementation of the IPsec protocol suite running on the Linux operating system. IABG has developed a specific patch what enables IPv6 and the use of certificates.

Both implementations are interoperable between them and they support PKIv6 certificates.

End-To-End web solutions with SSL using certificates

In the case of web solutions, we have to distinguish between web clients and web servers. Web clients are usually browsers, whereas web servers are HTTP servers, therefore both applications are totally different.

The main open-source web servers supporting SSL are Apache and Tomcat. Both solutions are projects of Apache Software Foundation.

- The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server. Apache is the most popular web server on the Internet. The Apache HTTP Server module mod_ssl provides an interface to the OpenSSL library, which provides SSL protocol. The webmaster can define the SSL configuration and identify valid users of a web application using client certificates.
- The Jakarta Project creates and maintains open source solutions on the Java platform for distribution to the public at no charge. Jakarta products are developed and distributed through various subprojects; Tomcat is one of these subprojects. Tomcat is the servlet official container for the Java and JavaServer Pages technologies. Tomcat support SSL for default. The webmaster must define a Tomcat module called Realm to identify valid users of a web application using client certificates.

The main web browser supporting SSL are Netscape Navigator and Microsoft Internet Explorer. Both solutions can be freely downloaded. When a browser wants to establish an SSL connection with a web server requiring client authentication, it needs to be able to access the client certificate. There is no single standard method for doing this. The two main browsers use different methods to store/retrieve certificates locally and externally. Both browsers use locally independent certificates and key databases. To store externally a client key and/or certificate, the main technology is smartcard.

One of the main aims of this task has been to use the public key capabilities of the smart card to enable cardholders to establish an SSL connection with a web server requiring client authentication. To do so, the browser needs to be able to access the data and functions on the card. There is no single standard method for doing this. Netscape Navigator uses PKCS#11, also known as Cryptoki, from RSA laboratories, while Microsoft Internet Explorer uses CryptoAPI, which is Microsoft's own definition. These two methods have some similarities, and both allow a service provider module in the form of a DLL to be "plugged in" to the browser. This DLL is called Cryptographic Service Provider (CSP) for Microsoft and PKCS #11 module for Netscape.

Smartcard support for PKIv6

PKIv6 supports use of smart cards, in RA and CA entities and, also, in web environments. Users will be able to have smart cards to store their cryptographic information. Each smart card will contain the user's private key, associate certificate and CA certificate. If user has a smart card, it can be used to store private and public key thanks to the PISCIS Cryptographic Service Provider.

Design solutions for project services with end-to-end security requirements

We have evaluated the project services ISABEL, AGWS, NAGIOS and VPN Enforcement Tool and have designed and developed solutions for their end-to-end security requirements.

- **ISABEL**. There are two connections for an ISABEL terminal, one is between the terminal and session master, and the other is between the terminal and flowserver. Both connections are insecure. A VPN solution using terminal certificates provides end-to-end security in these connections. Because ISABEL runs over Linux, FreeS/WAN is the appropriated VPN solution.
- **AGWS**. This system needs to identify the valid users. AGWS runs over Apache and the users access it through a web browser. To get end-to-end security in this project application, SSL configuration forces users to have a certificate and AGWS uses the client certificates to identify the valid users.
- **NAGIOS**. This web tool must identify the administrators. Now this identification is possible with login and password, but this method is not strong secure. SSL with client authentication using certificates gets end-to-end strong security and enables identify the valid administrators.
- **VPN Enforcement Tool**. This tool must identify the valid VPN administrators before the administrator can access the VPN configuration. The VPN administrator uses a web browser and VPN Enforcement Tool is located in a web server Tomcat. To get end-to-end security in this project service, we have to develop a specific module Realm and generate certificates with a specific OU for each VPN administrator.

Future Work

- Continue the study and analysis of the End-To-End solutions.

2.1.3 Security Policy

Partners

UMU

Objectives

- Continue the development and evaluation of VPN Enforcement Point.
- Evaluate the policy-related configuration protocols and select the most appropriate protocol (or set of protocols).
- Design an internal architecture for the policy framework components (PDP, PEP, and PMT).
- Design a generic, secure, scalable, and distributed policy framework.
- Define and design an IX environment for the provisioning of a dynamic VPN service.
- Analyze and design the integration of project services and applications with the designed policy framework.

Description

Manual IPsec policy configuration is inefficient and error-prone. An erroneous policy could lead to communication blockade or serious security breach. This is one of the main motivations for the appropriateness of using dynamic secure VPNs. Just one real example could be that manual IPsec configuration of a VPN between two or more secure gateways is not added at the same time in all the gateways, which can lead to a temporary inconsistent network situation. This can be easily solved using dynamic secure VPNs, where all the configurations can be done simultaneously. So a logic extension to secure VPN is the use of dynamic secure VPNs. We present dynamic IPv6 VPNs based on the development of a PKIv6 infrastructure and a Policy Management System based on IETF/DMTF policy framework.

The VPN provisioning is an added-value service with important advantages against the static VPN definition and an area of great impact in the real deployment of security services over IPv6 and the future Internet.

Relevance

Policy-based VPN management systems unify IPsec configuration parameters to provide consistent security behavior across the organization, regardless of the type and/or number of devices involved. Thus, it provides scalability and flexibility to the management of network and services.

Results

The dynamic VPN provisioning is based on the definition of a set of policies automating VPN creation, and a set of components that maintain the VPN description and on which IPsec nodes to apply policies. The creation of secure VPN through dynamic VPN provisioning has, among others, the following advantages:

- Configure and manage automatically and systematically various IPsec policies in a user-friendly way.
- Easier management in medium and large-scale networks.
- Avoid interoperability problems between different implementations.

- Network administrators do not need to configure every individual device.
- It allows an easier, faster, and more scalable deployment of VPNs.
- It provides a holistic view of the network from the network manager's perspective.

A policy management system is, therefore, demanded to provide dynamic VPN provisioning. The concept of management policy is related to one or more rules, which in turn are decomposed in conditions and actions. Whenever the condition(s) is fulfilled, the action(s) is executed. The scope of the set of rules that will drive the management of a system or service can be defined in any application domain and at any functional level. This paradigm is generic and valid for other application environments as such access control, QoS, etc.

We have developed one policy management system based on IETF/DMTF policy framework, named UMU Policy-Based Network Management (UMU-PBNM). It consists of four main components for policy control: The Policy Management Tool (PMT) that is user interface to allow constructing policies, deploying policies and monitoring the status of the policy-managed environment; the Policy Repository is used to store the policies generated by the management tool; the Policy Enforcement Point (PEP) that is a component running on a network node that can apply and execute the different policies; and the Policy Decision Point (PDP) that has a view over the whole network area to control and it is responsible for interpreting the policies stored in the repository and communicating them to the PEP.

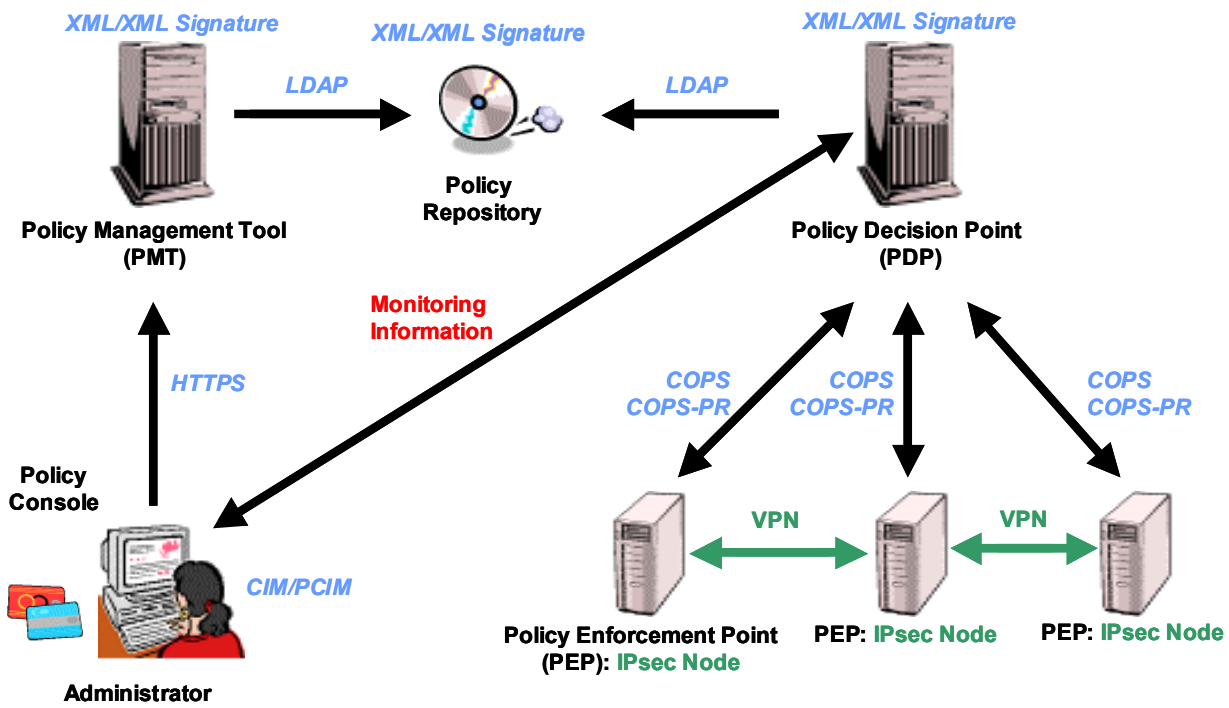

Figure 2-5: UMU Policy-Based Network Management

In UMU-PBNM, the whole policy life cycle (definition, storing, distribution, processing and enforcement) makes use of a XML technologies (e.g. XML schemas, XSLT, XML encryption, XML signature, etc.) and tools.

The most important features for this new proposal are:

- The use of UMU-PKIv6 to define, manage and enforce secure policy documents. This feature is based on the use of the XML Signature and XML Encryption standards.
- UMU-PKIv6 is also used to authenticate all the elements (PMTs, PDPs, PEPs) managing, storing and exchanging policies.

- The design and implementation of easy-to-use web-based assistants to help network managers, who in some cases may not be very familiar with XML technologies and digital signatures, to create XML signed policies according to one particular XML schema.
- The use of COPS and COPS-PR protocols with a new extension to carry XML documents.
- The definition of policy-based management architecture independent of the service and/or application being managed.
- Flexibility, as it is easy to add new nodes to be managed in run-time. It is also easy to add new networking areas to the proposed architecture

UMU-PBNM has four main phases: definition, distribution, enforcement, and monitoring. Next sections provide a detailed explanation of each phase, and how XML is integrated in them.

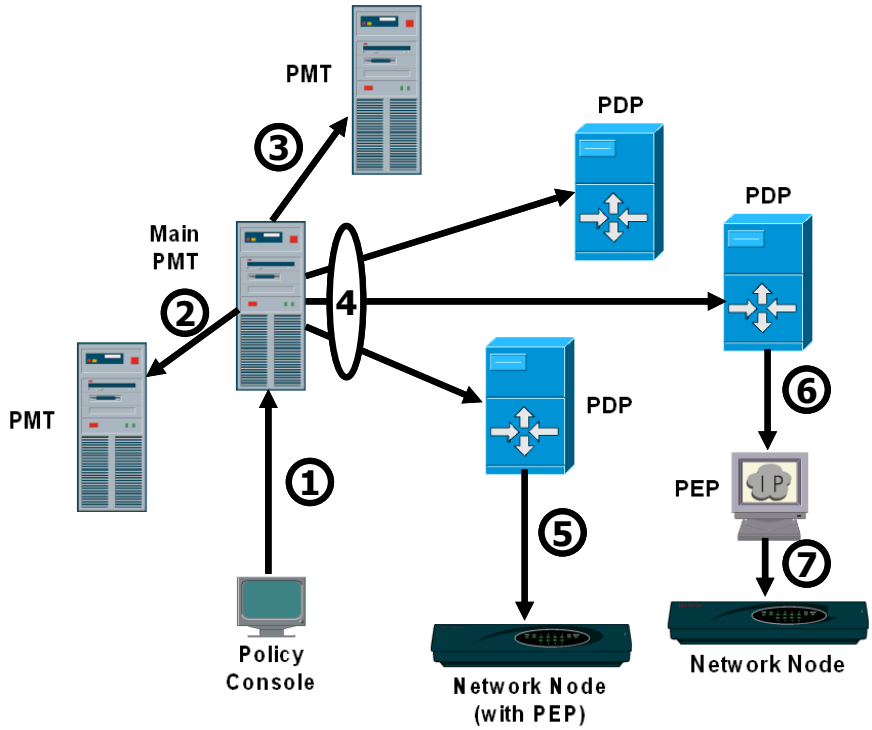

Figure 2-6: The Policy Management Process in the UMU-PBNM Architecture

Phase 1: Defining Policies

In the first phase of the UMU-PBNM, the administrator interacts, using his policy console with one of the policy management tools (PMT) existing in the system. This allows him to create new policies for one specific service and/or application, or edit or remove the existing ones (message 1).

In this phase, the communication protocol in use is HTTPS (i.e. HTTP + SSL), which provides confidentiality, integrity, and mutual authentication between the policy console and the PMT server. To proceed, the administrator has to enter the secure URL (HTTPS) of one of the policy management tools existing in the system, and provide access to his private key and certificate, which could be stored in an encrypted file in his PC or in his smart card.

Once this basic security check is done, the PMT will provide the manager access to this set of functionalities:

- Create a new policy.
- Modify an existing policy.
- Delete an existing policy.
- Check the monitoring information (warnings, errors, policy conflict information, etc.) generated by the policy management system.

The administrator can define a new XML policy document, and then the policy is signed and sends to the PMT server where it is stored in a native XML database, and then distributed to other PMTs and PDPs servers existing in the system. This distribution process defines the second phase of our architecture.

Phase 2: Distributing Policies

Every time one of the system managers create, modify or delete one policy document, the specific policy management tool server receiving it, proceeds to send it to any other PMT server existing in the management architecture (as indicated by messages 2 and 3). It provides fault-tolerance to the proposed architecture.

Phase 3: Enforcing Policies

Once a policy decision point (PDP) receives a new policy information (because a new document has been created, or an existing one has been modified or deleted), it has to decide if this policy needs to be applied or not to each of the policy enforcement points (PEPs) connected to it. Moreover, when a new PEP is active in the network it needs to get or update the internal configuration it is using. In both cases, a communication needs to be established between both elements. One of the most significant protocols specified by the IETF to do this task is COPS (Common Open Policy Service protocol), and its extension for provisioning policy models, COPS-PR (COPS Usage for Policy Provisioning).

UMU-PBNM has defined an extension to the COPS and COPS-PR protocols for supporting the exchange of XML documents between the PDP and the PEP. It allows the PEP to report device capabilities to the PDP using XML, and to the PDP to provide XML-based configuration information and updates to the PEP. This extension has been successfully tested with both the Vovida 1.5 COPS implementation and with the UMU-jCOPS (University of Murcia Java COPS) implementation used in the UMU-PBNM.

Once the policy information defined in XML is in the PEP (message 5 and 6), it normally has to convert it to the specific configuration format used by the device that it is controlling. Then it will apply it (message 7). To do this, another XML technology is used: XSLT (eXtensible Stylesheet Language Transformation). With it, we convert from device-independent XML policy documents into specific configuration files.

Phase 4: Monitoring Policies

So far, the have described the definition, distribution and enforcement of policies, and how XML technologies can be integrated into these management processes. But there is one more aspect of policy-based network, service and application management that needs to be considered: monitoring.

In the case of the UMU-PBNM, it has defined the monitoring process depicted in the following figure.

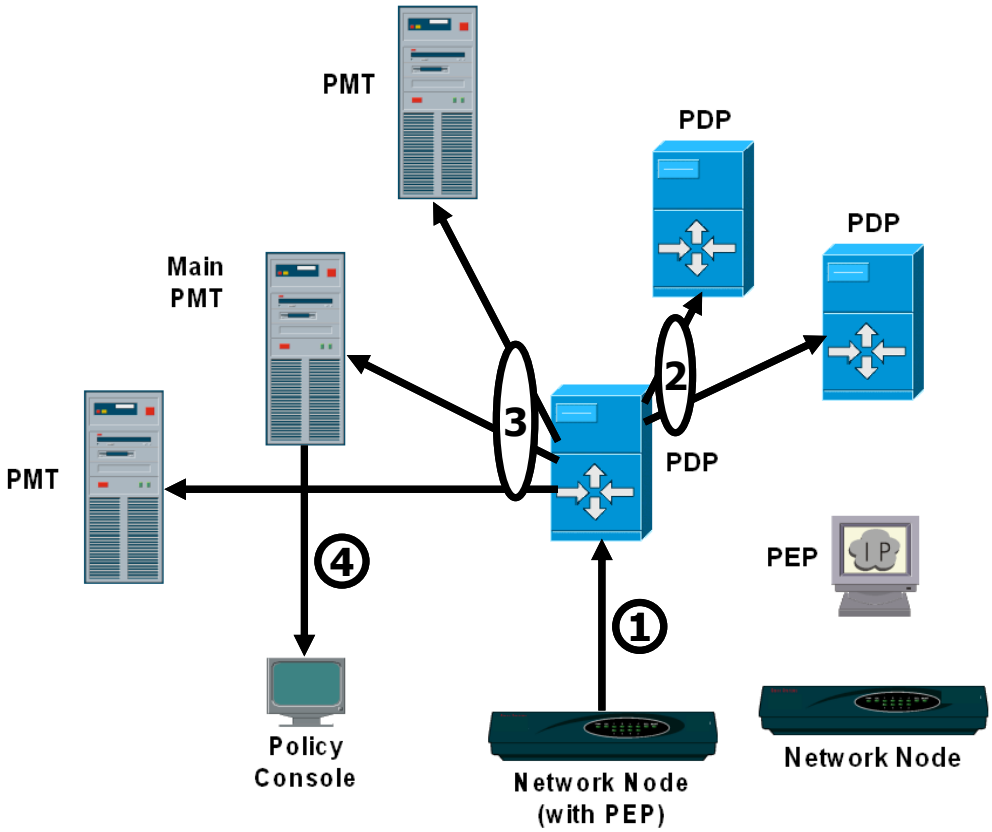

Figure 2-7: The Policy Monitoring Process in the UMU-PBNM Architecture

Message 1 shows a COPS interaction between a network node and the PDP where it is registered. Messages 2 and 3 shows the same interaction, but with a non-COPS network node. Monitoring information is defined as XML signed documents, based on one particular XML schema.

Once the monitoring information is in one PDP, it distributes it to any other PDPs (for fault-tolerance reasons), and to the PMT servers (messages 4, 5, and 6). The last message, number 7, in this figure describes the interaction of an authenticated administrator accessing securely from his policy console to one of the PMT servers available in the infrastructure. This allows him to see how the network, services, applications, and their related policies, are behaving, and to act consequently.

Future Work

- Cisco support for VPN Enforcement Point.
- Definition and design of an IX environment for the provisioning of a dynamic VPN service.

2.1.4 IPv6 Filters and Firewall Functions

Partners

6WIND

Objectives

The goal of this activity is to define, implement and evaluate the IPv6 filters and the associated tools that would be of interest in Euro6IX environment. The initial study should also identify IPv6 specific vulnerabilities.

Description

IP filters are appropriate mechanisms when a private network must be protected against insecure traffic from the outside. IP filters are part of the security solution when dealing with a network such as the one in the following picture.

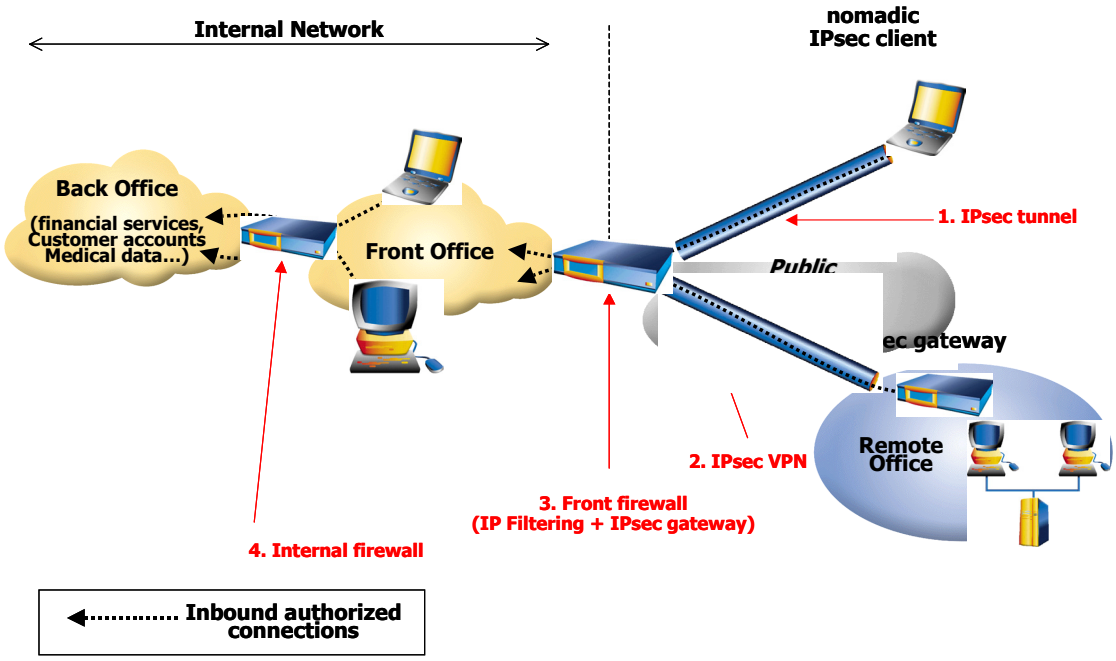

Figure 2-8: IPv6 Filters and Firewall Functions (I)

It is generally advised to use dynamic ("stateful") packet filtering which keeps a state on opened sessions. This technology is now mature and more secure than static filtering.

Up to now, IPv6 filters are an area not well studied. IPv6 implementations are not always as complete as IPv4 ones from a security point of view. For instance, stateful filtering is not always supported, even in very well known implementations. And even when it is, its impact on performance should be measured.

The need for IP filters was clearly identified in WP2's specifications. 6WIND has started the state of the art and study of IX requirements. Besides, related to this activity, 6WIND has also carried a few experiments. It has protected its local sites with IPv6 filters (WP3) and also installed a lightweight-monitoring tool.

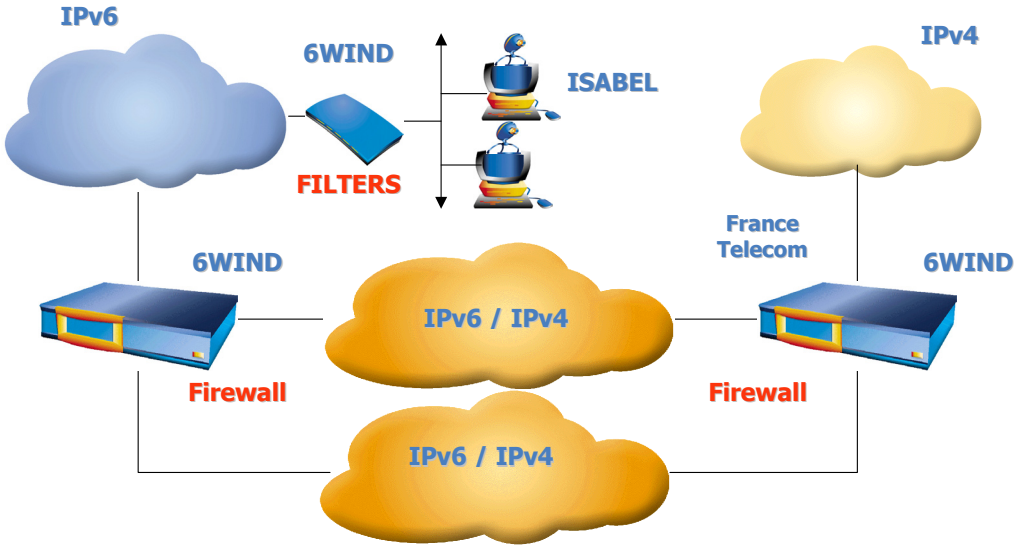

Figure 2-9: IPv6 Filters and Firewall Functions (II)

In conjunction with the mobility activity, 6WIND has also developed a prototype in which a Radius server dynamically controls IPv6 filters.

Results

The IPv6 capabilities of the most well known firewalls have been followed-up. This includes commercial firewalls as well as open-source firewalls and network filters. The comparison between stateless and stateful filtering has been done and the results have been provided to the D4.4 document. It is important to remind and describe the features of each type of filtering since the firewall is a key component used to build security architectures required by ISP.

As the 6WIND SixOS operating system includes all the features required for an IPv6 network firewall, a small presentation of its possibilities have been done as a contribution which have been provided to the security report D4.4. The SixOS provides a range of features that include Differentiated Services (DiffServ) Quality of Service (QoS) management, Multicast, IP mobility, Security mechanisms based on IP Security (IPsec), IKE and IP stateful Firewall. The features are available for both IPv4 and IPv6 IP standards and built into 6WIND's equipment: 6WINDGate 6200 Series and 6100 Series. The Security features allow the 6WINDGates to be compared to commercial or open-source IPv6 firewalls.

Relevance

IPv6 networks will need filters and firewalls like it is the case for IPv4 networks. Therefore it is important to verify the IPv6 capabilities and specificities of those security tools. These kinds of tools were essential in IPv4 networks. It will also be the case not only during the transition period when IPv4 and IPv6 are co-existing but also for IPv6 only networks.

Future Work

- Finish the state of the art and study of IX requirements.
- Complement TID document.
- Study the impact in IX.
- Continue the experimentation of IPv6 access control.

2.1.5 Security of IPv6 Routing Protocols

Partners

6WIND

Objectives

The goal of this activity is to study, test and implement secure IPv6 routing protocols, either inside autonomous systems (RIPng, OSPFv3) or between them (BGP). The current specifications cannot be implemented easily, as they mandate the use of IPsec. In the case of BGP, the main goal is to check if current BGP security mechanisms are sufficient.

Description

A network such as the Euro6IX must use secured routing protocols. 6WIND has established the state of the art and we have studied IPsec solutions. Currently, the proposed solutions are:

- **Securing RIPng with IPsec.** The goal is to get at least the same level of security as the MD5 authentication of RIPv2 (RFC 2453). At the level of the IPsec stack, a static security association (SA) is linked to security policies (SP) and this SA is used regardless of addresses (wildcard SA). SPs describe RIPng flows: Link-local toward to RIPng link-local multicast group or to on-link neighbors, UDP, source and destination ports 521.
- **Securing OSPFv3 with IPsec.** The proposed method for securing OSPFv3 (RFC 2740) consists in using IPsec AH in transport mode, with a minimal configuration. At the level of IPsec stack, a static security association (SA) is linked to security policies (SP) and used regardless of addresses (wildcard SA). SPs describe OSPFv3 flow: Protocol 89. The method is compatible with the proposition given in draft-ietf-ospf-ospfv3-auth-01.txt.

Results

The results that are described hereafter represent a 6WIND development that has been implemented in prototypes of the 6WIND router. It is currently a rather proprietary implementation since the IETF specifications are not yet stable. Several solutions are discussed at the moment; however it is roughly in accordance with works currently realized in the IETF workgroups.

For both protocols RIPng and OSPFv3 an internal test-bed has been setup to validate and test the prototype implementation. The feature including routing and security is tested between two 6WIND prototypes (Figure 2-10).

Each router has been configured with:

- Static SAs, reserving a SPI for RIPng traffic:SPI = 909090.
- AH transport mode.
- HMAC-MD5.
- Interface index.
- Policy index reference.

The security policy index binds static SA to RIPng policies:

- Selectors: Any to fe02::9 UDP port 521, interface index.
- Policy index.
- Action: IPsec required.

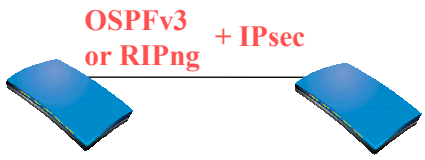

Figure 2-10: RIPng or OSPFv3 + IPsec Scenario

Thanks to IPsec requirement on inbound policies, clear RIPng control traffic is not accepted from routers, and outbound RIPng traffic is authenticated (AH transport) by mean of a special look-up in Security Association database on a static SA that matches policy index. Using interface index, a per-interface secret can be defined.

As it has been tested, two 6WIND routers configured with the previous security rules exchange correctly IPv6 routes. The exchanged packets format looks like:

- IPv6 header: fe80::X to fe02::9, Next Header = AH.
- AH header: SPI = 0x909090, Next Header = UDP.
- UDP header: Destination port 521.
- RIPng data.

In the case of OSPFv4, the IPsec configuration is done in a quite similar way. The main difference is the protocol selector: while it is UDP port 521 for RIPng, it is protocol 89 for OSPFv3. Also two IPv6 multicast addresses are used by OSPFv3: All-OSPF-Routers (ff02::5) and All-Designated-Routers (ff02::6).

We have not addressed the case when virtual links are used. The Security SA is:

- SPI = 333333.
- AH mode transport.
- HMAC-MD5.
- Interface index.
- Policy index reference.

The security policy index binds static SA to OSPFv3 policies:

- Selectors: any to (ff02::5, ff02::6) protocol 89, interface index.
- Policy index.
- Action: IPsec required.

As it has been tested, two 6WIND routers configured with the previous security rules exchange correctly IPv6 routes. The exchanged packets format looks like:

- IPv6 header: fe80::X to fe02::5, Next Header = AH.
- AH header: SPI = 0x333333, Next Header = OSPFv3.
- OSPFv3 data.

Future Work

- Future work will concentrate on examining how these implementations of secure routing protocols can be used inside the Euro6IX routing architecture.

2.1.6 Network Security

Partners

TID

Objectives

The objective of this activity is to study and contribute to the design of security policies for Euro6IX network. Information security must be preserved according to three basic components:

- Confidentiality: to assure that only authorized users or systems have access to different types of information.
- Integrity: to provide guarantees about accuracy of information.
- Availability: to have access to the information when required by appropriate users.

Description

This activity started in Y1 with a description and analysis of network security elements (filters and firewalls) and the presentation of a first approach to a network security model applicable in the Euro6IX context.

Along 1Y2, a generalized network model applicable in the Euro6IX context has been defined, interfaces between functional blocks have been identified, and network security measures both at these interfaces and at network elements to protect the network from attacks have been recommended. Apart from defining the network security, all these measures have been implemented in the network during 2Y2.

Results

Current result is a document that presents several security measures. The basic security feature to consider is the Least Privilege principle. It states that every component should have only the privileges needed to carry out assigned tasks. Taking this rule as a reference, the main security characteristics treated in this document are easily identified looking at its table of contents:

1. The basic network model
2. Traffic control
    a. Basic rules for traffic filters
    b. Traffic control in the IPv6 Internet Exchange
    c. Traffic control in the carrier's network
    d. Traffic control in the ISP network.
3. General security features
    a. Time synchronization
    b. Backups
    c. Logs management
    d. Intrusion detection
    e. Periodic checks of security measures
4. Security measures at network elements
    a. General concerns
    b. Network configuration measures
    c. Logs management
5. Security measures for hosts
    a. Local security

   b. Network security
   c. Access
6. Services running in hosts
7. Specific recommendations for ISPs.

1. The basic network model

The Euro6IX network model is divided into the following functional blocks:

- **Internet Exchange Network**.
- **Carrier's network**. It can be subdivided into the following blocks: Provider backbone, point of presence (POP), network service centre and network management centre.
- **ISP network**.

2. Traffic control

This section starts with basic rules for traffic filters. Then, for each part of the network model a traffic model, which establishes different levels of security is defined. Then, each one of these levels is studied and security recommendations are stated.

a. Basic rules for traffic filters

Summarizing, these rules identifies the best location of filters depending on the network element to protect. They also remind the importance of not duplicating filters to not disturb network performance.

b. Traffic control in the carrier's network

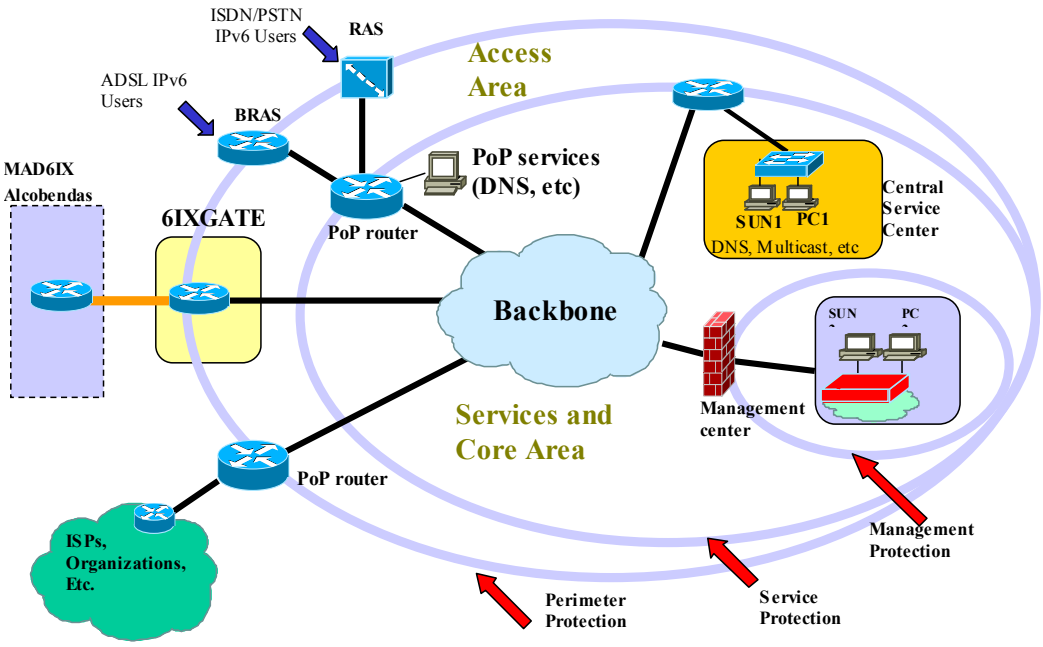

Figure 2-11: Traffic Control in the Carrier's Network

c. Traffic control in the ISP's network

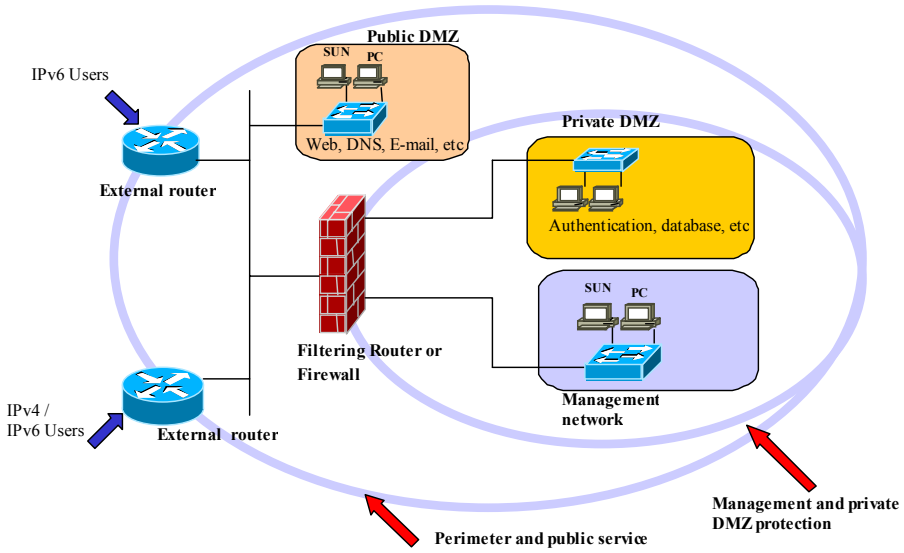

Figure 2-12: Traffic Control in the ISP's Network

d. Traffic control in the IPv6 Internet Exchange

In the following figure, traffic model for Internet Exchange is showed. In this case, only one circle is depicted, but there are two protection levels: Perimetral and service/management protection. The reason is that the same network element carries out perimetral protection and, depending on which network is connected to, also implement service or management protection. Each protection level is configured with traffic filters located at different interfaces.

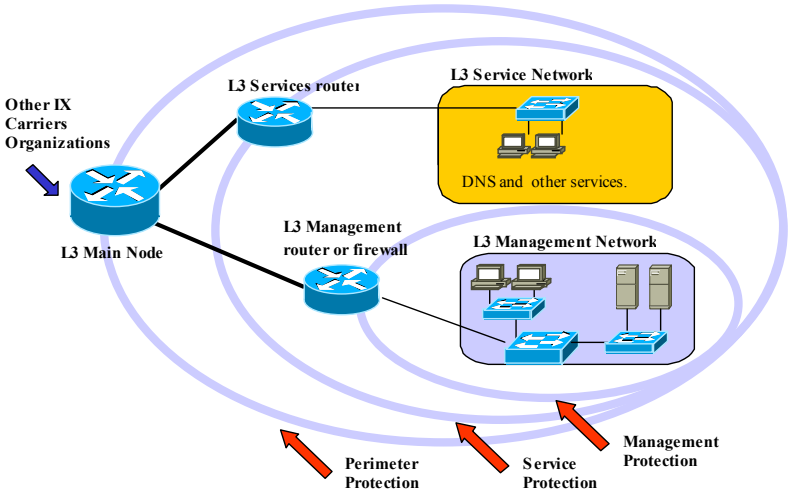

Figure 2-13: Traffic Control in the IPv6 Internet Exchange

Note: From a security point of view and if performance it is not seriously affected, it is recommended that traffic flows in the IPv6 Internet Exchange from/to L3 Services and Management networks pass through its access element (like filtering routers/firewalls) which provide connection for this parts of the network. For example, traffic from/to L3 Management Network should pass through the filtering element that connects it to the rest of the IPv6 Internet Exchange; the same it is applicable to L3 Services Network.

3. General security features

This section refers to some general measures such as time synchronization, backups, logs management intrusion detection, to aid network protection.

The main reason to make specific security configurations for hosts is that in a default installation of an operating system, operation is the most important aspect, not security. So, many services and daemons are running and most of them are not needed and this open security holes. Also, different machines hosting different services require dedicated configurations.

This section presents measures to increase host security. Due to the fact that hosts of different types can be present in a network, which permit different things to implement, additional measures can be taken depending on the particular system treated.

The same problem described for security for hosts is applicable to services. Default installations are not so secure as desired. So important as harden operating system configuration (which has been presented in above subsections) is to secure applications/services that are running in network hosts. Web, DNS or E-mail services should be configured properly to increase its security. Depending on the service and vendor, specific measures must be taken. As a general recommendation for most services, concerns related to access control, activation of application logs that services can generate, etc. should be considered.

4. Specific recommendations for ISPs

In this section some security measures to consider for ISPs networks are presented. It can be resumed in the following points:

- Whenever is possible, different services should be run on different machines. This feature can aid to limit the impact of a security hole exploited on one service. Additionally, this allows making a more restrictive configuration for each service (for example each server can be configured to accept only the connections it requires to give its service).
- Also, DMZ LANs should be formed with switches to avoid that traffic directed to one service affect other servers.
- Servers should not be attached to transit network segments.
- ISPs should take necessary actions to prevent their mail infrastructure from spamming. Black Lists of mail servers available on Internet should be considered. Also they must advise to their customers to take steps to prevent this activity in their own systems.
- Message submissions should be authenticated using AUTH SMTP service extension as described in RFC 2554.
- ISPs should have an appropriate use policy, including measures to be in accordance with Data Protection Legislation (if exists in its jurisdiction).
- ISPs should consider procedures to inform their customers, by means of a secure channel (if possible), about security incidents and notify them about vulnerabilities in the services they provide. Also ISP must implement an appropriate method to receive and handle incidents from their customers.

For further information RFC 3013 can be consulted.

5. Security implementation at TID IPv6 network

At this time some of the features described above are being implemented at TID IPv6 network. The work done up to now is focused in traffic control and securing network elements (Cisco routers).

a. Traffic control

For traffic control, filtering has been defined for carrier, ISP and IX networks.

Filters for Cisco routers located at carrier network PoPs has been configured, applied to interfaces and tested. These include interfaces with external clients and service interfaces.

The router connecting with Internet Exchange has filters configured but not applied nor tested yet. These filters are for management, service and external access interfaces.

Filters previously defined for Network Management and Service Centers are going to be modified due to changes in network implementation (there are availability of new routers that will be included in the network)

Filters for ISP network has been configured and partially tested. At this time they are not applied so it is not possible to complete the tests by now to avoid interferences with other important tasks in the network.

For Internet Exchange network, filters have not been configured yet, for the same reason that above.

Note that filtering is continuously been studied and modified as network implementation changes (network connections, new routers or IOS versions are included, services or accesses are moved to other locations in the network, etc).

b. Securing network elements

Regarding to securing network elements, features allowed by available IOS nowadays has been configured and tested for carrier and ISP routers.

These characteristics can be divided in three categories: general, router access and interface-specific.

General parameters include password encryption, disable unnecessary services (such as finger or HTTP), configuration of logs, disable CDP (Cisco Discovery Protocol), configuration of access banners, disable source routing options.

Router access options are console, aux and VTY configuration, including passwords, inactive session timers and access lists.

On the other hand, interfaces security features are options such as disable IP Proxy-ARP and IPv6 redirects (if needed).

Note: It is pending to configure NTP (Network Time Protocol) and SNMP (Simple Network Management Protocol) for IPv6.

2.1.7 Intrusion Detection Systems

Partners

TID, Telscom, 6WIND

Objectives

- To identify attack detection schemes.
- Detect what advantages/disadvantages offers IPv6.
- New schemes introduced by IPv6.
- To develop small apps to simulate attacks and test the IDS.
- To provide the IDS logic detection study and testing to the developers.

Description and Results

1. Introduction to Intrusion Detection Systems

IDSs detect attacks against a system by monitoring network traffic and looking for well-known pattern attacks or anomalies in protocol interchanges.

The Network IDSs (NIDS) monitor network traffic from all machines, whereas Host IDSs (HIDS) monitor only one-machine parameters (log files, CPU rate, etc).

2. IPv4 Network Intrusion Detection Systems

Actually, the most important IPv4 NIDSs are:

- Dragon.
- Network Flight Recorder.
- Cisco IDS.
- Snort.
- RealSecure.

3. NIDS Development in an IPv6 network

3.1 NIDS general development guide

3.1.1 Probably intrusion sources

These are:

- Externally, all other national organizations networks, international connections through the interchanger, universities that access the network, TID intranet and PSTN/RDSI/ADSL users.
- Internally, incorrect use of resources by administrators and operators.

3.1.2 NIDS rules

- Forbidden accesses:
    - Telnet accesses rules: Login failed detection, telnet accesses login as *root* detection, general telnet accesses detection and brute force attacks detection.

  - FTP accesses rules: Login failed detection, FTP accesses login as *root* detection, General FTP accesses detection and brute force attacks.
  - Port scans rules: TCP connection detection, *SYN FIN* scan, *NULL* scan, *XMAS* scan and nmap *XMAS* scan.
- Common IP attacks:
  - Critical services (finger, systat, ttymux, netstat, ...) scans detection.
  - portmap access and RPC services attacks (ttdbservd, cmsd, statd, mountd, sadmin, snmpXdmid, ...) detection.
  - Shell executions detection.
  - DoS attacks detection.
  - *Land Attack* detection.
  - *IP sppofing* detection.
  - FTP specific attacks: CWD ~root command detection and *passwd* file getting by FTP detection.
- Web servers attacks:
  - Vulnerable CGI scan using *Nessus* detection.
  - CGI specific attacks detection.
  - Shell executions detection.
  - *passwd* and *shadow* files accesses detection.
- DNS servers attacks:
  - TCP transfers from forbidden IP addresses detection.
  - BIND version-getting detection.
  - IQUERY (inverse query) detection.
- e-mail servers attacks:
  - e-mail server scans using EXPN and VRFY commands detection.
  - sendmail server exploits detection.
  - Corrupt MAIL and RCPT commands sent to SMTP server detection.

3.1.3 Response strategies

It's highly recommended that a NIDS responses to attacks by sending SNMP traps or an e-mail to network administrator. TCP resets are useful too.

3.1.4 Sensors installation

Recommendations:

- When sensors are connected to switches, the port where it's connected to must be configured in SPAN mode.
- Traffic may be balanced between different sensors. It also may be sampled.
- To improve security, sensor-monitoring interface shouldn't have IP address. But a second interface with IP address should be necessary to manage sensors.

3.2 NIDS Development at Access Centers

Access Centers (POPs) are accessed by users, Internet Service Providers and universities, so it's a security critical point.

A sensor must be placed in every POP of the network only to monitor DNS and similar services traffic.

Rules about forbidden accesses and common attacks are needed. Specific rules about services in POP (like DNS) are useful too.

3.3 NIDS Development at Interchangers

Interchanger is the point where national and international network companies connect, in addition to owner Telco customers and some universities. Interchanger also connects within other international interchangers.

A sensor must be placed in the IX service network to detect attacks against servers. Another sensor must be connected to the IX management network to detect attacks against management equipments.

Rules about forbidden accesses and common attacks are needed. Specific rules about services in the IX service network (like DNS) are useful too. Also, specific rules about management servers in the IX management network should be developed.

3.4 NIDS Development at ISP

ISP is a high vulnerable part of the network, because of clients and users accessing from Internet, 6Bone and Intranets.

A sensor must be placed in the ISP management network. A second sensor should protect ISP services, and it should be placed in the DMZ.

Rules about forbidden accesses and common attacks are needed. Specific rules about services in the ISP service network (like DNS, Web, Mail, ...) are useful too.

3.5 NIDS Development at Backbone

Backbone is acceded by POPs and Interchanger, so it's the most vulnerable part of the network. In addition, Central Management Center is connected to backbone, so an attack against it would involve the entire network.

A sensor must be placed between Backbone and Interchanger, but it's necessary to assure that the sensor can monitor a very high traffic rate.

A second sensor must be placed in Central Management Center, and a third sensor in Central Services Center. Note that an attack to Web server in Central Services Center involves that Central Management Center become insecure.

Rules about forbidden accesses and common attacks are needed. Specific rules about Web servers are useful too.

3.6 An IDS tool IPv6 Enabled: SNORT

An Intrusion Detection System (IDS) looks for attacks. In particular, it searches for specific patterns that usually indicate malicious or suspicious intent.

Snort is a software program that can be configured as sniffer, packet logger, and as network intrusion detection system. To enable network intrusion detection (NIDS) mode the syntax is:

./snort -dev -l ./log -h <IP address range> -c snort.conf

Snort.conf is the name of the rules file. Each packet will follow the rules set in the snort.conf file in order to decide whether an action should be taken.

The scenario in which this package was tested is represented in the following picture:

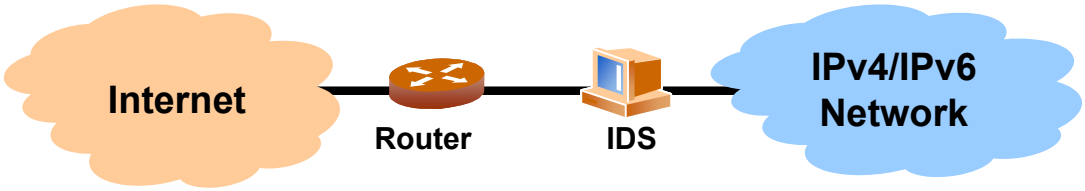

Figure 2-14: IDS Scenario

The summary log is shown in the following picture:

```
===============================================================================
Snort analyzed 21442 out of 21442 packets, dropping 0(0.000%) packets

Breakdown by protocol:                    Action Stats:
    TCP: 2915        (13.595%)            ALERTS: 137
    UDP: 3134        (14.616%)            LOGGED: 137
   ICMP: 77          (0.359%)             PASSED: 0
    ARP: 5657        (26.383%)
  EAPOL: 0           (0.000%)
   IPv6: 9401        (43.844%)
    IPX: 0           (0.000%)
  OTHER: 3           (0.014%)
DISCARD: 0           (0.000%)
===============================================================================
Wireless Stats:
Breakdown by type:
    Management Packets: 0          (0.000%)
    Control Packets:    0          (0.000%)
    Data Packets:       0          (0.000%)
===============================================================================
Fragmentation Stats:
Fragmented IP Packets: 0          (0.000%)
    Fragment Trackers: 0
   Rebuilt IP Packets: 0
   Frag elements used: 0
Discarded(incomplete): 0
   Discarded(timeout): 0
  Frag2 memory faults: 0
===============================================================================
TCP Stream Reassembly Stats:
        TCP Packets Used: 2906       (13.553%)
         Stream Trackers: 247
          Stream flushes: 12
           Segments used: 23
   Stream4 Memory Faults: 0
===============================================================================
```

Figure 2-15: SNORT Summary Log

The SNORT IDS has been installed in a test-bed and tested in order to evaluate its IPv6 capabilities. The result is that the version under test, 2.0.0, has still very few IPv6 capabilities compared to the possibilities provided in IPv4. This applies especially concerning the analysis and decoding of IPv6 packets, which is already possible with genuine packet analysers, ethereal and tcpdump.

Relevance

IPv6 networks will need IDS like it is the case for IPv4 networks. Therefore it is important to verify the IPv6 capabilities and specificities of those security tools. IPv6 IDS tools will not avoid more attacks in IPv6 than in IPv4 but there are nevertheless essential.

2.1.8 Transition Mechanisms and Security

Partners

6WIND

Objectives

The objective is to identify and describe specific vulnerabilities that could be introduced with the implementation of IPv6, especially when interconnecting the new generation networks with the pre-existing IPv4 networks. Once the vulnerabilities are identified, the work consists in finding and defining the security solutions in order to improve the security level in exchange network environment that could mix IPv6 and IPv4 networks.

Description

There will be no D-day for migrating from IPv4 to IPv6 networks. Therefore new IPv6 networks will have to cohabit during a long period with the IPv4 networks. Several transition mechanisms have been designed to make the transition as transparent as possible. Nevertheless, new security threats are emerging due to the existence of the transition period. Therefore new security solutions have to be deployed in order to keep an appropriate security level.

IPv6/IPv4 transition mechanisms will introduce a new security challenge for ISPs. They must be aware of the vulnerabilities that could be introduced. In order to give them concrete information about this topic, some transition mechanisms, essentially mechanisms based on tunnels for the moment, have been studied. Whereas the manually configured tunnels are the less vulnerable, the automatically configured tunnels required to be completed by specific security measures. Generally, it can be noted that the transition mechanisms increase the possibility of Denial of Service (DoS) attacks. Tunnel broker and 6to4 are based on automatic tunneling which allows a hacker to encapsulate all its traffic while being anonymized by public automatic tunnel brokers and 6to4 routers. A 6to4 router may be used as a reflector and therefore being unwillingly involved in a DoS attacks. 6to4 vulnerabilities have been already detailed and described in IETF documents.

Relevance

Since new IPv6 networks will have to cohabit during a long period with the IPv4 networks, it is important to have a good idea of the security vulnerabilities involved by such architectures.

Results

The solutions are based on combinations of standard security mechanisms that only need to be ported to IPv6. In this context, IPv4/IPv6 filters still have a key role. In order to protect themselves against standard kinds of malicious traffic ISPs should implement some minimum-security configurations. Mainly, they should not allow blind packets decapsulations, the filters should perform ingress and egress filtering, martian addresses (encapsulated loopback, null, multicast or any kind of forged addresses) should be filtered. It is important to perform double filtering, i.e. before and after decapsulation, in order to be sure if the IPv6 addresses are confirmed by the IPv4 addresses.

The following paragraphs describe the concrete security measures that have to be applied in order to achieve an acceptable security level in a network using the 6to4 mechanism.

Double Filtering

Let's take the previous attacking scenario. In order to solve this problem, the ISP should apply the following rule:

```
Rule 30000 deny log 41 from any to any
Rule 31000 allow 41 from 9.0.0.2 to 8.0.0.1
```

By the default the ISP should deny any IPv4 traffic coming from the outside and containing IPv6 packets. When an enterprise wants to get connection to the IPv6 through the ISP's relays the administrator (manually or dynamically) will add the rule to allow the IPv4/IPv6 traffic coming from the enterprise CPE (Customer Premise Equipment).

The same rules should be applied on the IPv6 packets after the decaspulation (checking if 2002:v4addr and v4 match):

```
Rule6 30000 deny log all from any to any
Rule6 31000 allow log all from 2002:0900:0002::/48 to any
```

A packet would be forwarded only after passing those two filters (same rules should be applied for Manually Configured Tunnel).

Martian Address Filtering

On both side of the interface (inbound and outbound) we should perform strong martian address filtering, denying any IPv4 broadcast, multicast …

A simple example would be the following packet:

```
Src = 3ffe:ffff:5678::aaaa
Dst = 2002:0900:00ff::bbbb
```

After receiving the packet natively, if the relay doesn't check the destination address for subnet broadcast, it would send the encapsulated IP-IP packet to 9.0.0.255. All nodes in the subnet would receive this, and the responses would be directed to the relay.

To prevent possible attacks, tunnel interface should filters out and silently discard the following packets:

- Packets with IPv4 unspecified address as outer IPv4 source/destination (0.0.0.0/8).
- Packets with loopback address as outer IPv4 source/destination (127.0.0.0/8).
- Packets with IPv4 multicast address as outer IPv4 source/destination (224.0.0.0/4).
- Packets with limited broadcast address as outer IPv4 source/destination (255.0.0.0/8).
- Packets with subnet broadcast address as outer IPv4 source/destination.
- Packets with private address as outer IPv4 source/destination. (10.0.0.0/8; 172.16.0.0./12; 192.168.1680.0./16; 169.254.0.0./16).

The same kind of filtering should be applied on IPv6 addresses, specially:

- 0::/16 (compatible, mapped addresses, loopback, unspecified,…).
- fe80::/10 (link-local).
- fec0::/10 (site-local).
- ff02::/16 (link-local multicast).

Ingress/Egress Filtering

IPv4 over IPv6 tunnel interfaces are interface like any other, so installing usual ingress and egress filtering is an obligation. For example, the site should configure the inbound filtering rule to reject any source addresses that have been spoofed to belong to the site itself.

Consider a regular IPv6 ISP that has the 2001: ab::/24 prefix. The network operators should install filtering rules on its upstream interface like the ones suggested in the following examples:

- On input, to protect from someone sending packets from the Internet with this ISP's source address:

```
Rule6 29000 allow all from any to any
Rule6 30000 deny log all from 2001:ab/24 to any
```

- On output, to protect from someone in the site sending packet out to the Internet with a wrong source address (see RFC2827):

```
Rule6 29000 deny all from any to any
Rule6 30000 allow log from 2001:ab::/24 to any
```

Theft of Service

IPv6 ISP may want provide restricted services (6to4 relay or manual tunnel), therefore complementary security solutions need to be implemented in order to manage the restriction like it was done in IPv4.

Basic checks can be performed with the already mentioned filter functions, accepting only traffic coming from the clients' addresses. In addition, other access control mechanisms can be useful for account management and billing service (e.g. based on Radius).

IPsec

Other security holes are also present and unsolved by simple filtering policies, such as for example the following scenario:

Assuming our attacker is still also in the v4 domain, he sends this time the malicious traffic towards a 6to4 router. However, since the 6to4 tunnels are automatic tunnels that are not configured per pair, it should be noted that when communicating between native domains and 6to4 sites, the relays that will be used in the two directions are very likely different; routing is highly asymmetric. Thus, for this scenario it is not feasible to limit relays we accept traffic from with access lists.

This kind of vulnerability is not yet resolved. The only solution which can be possible is the authentication by AH IPsec. Each relay signs the traffic that it forwards to the native IPv4 backbone, thus we can recommend that the various ISP which wish to provide 6to4 connection services, circulate their relays public keys. So the 6to4 site administrators will then have criteria on which to base in order to be sure of the relay identity, which transmit the traffic. This solution could be the best answer to the problems of security in the Internet specially for this case, but it require a public key infrastructure to be deployed which is actually not available.

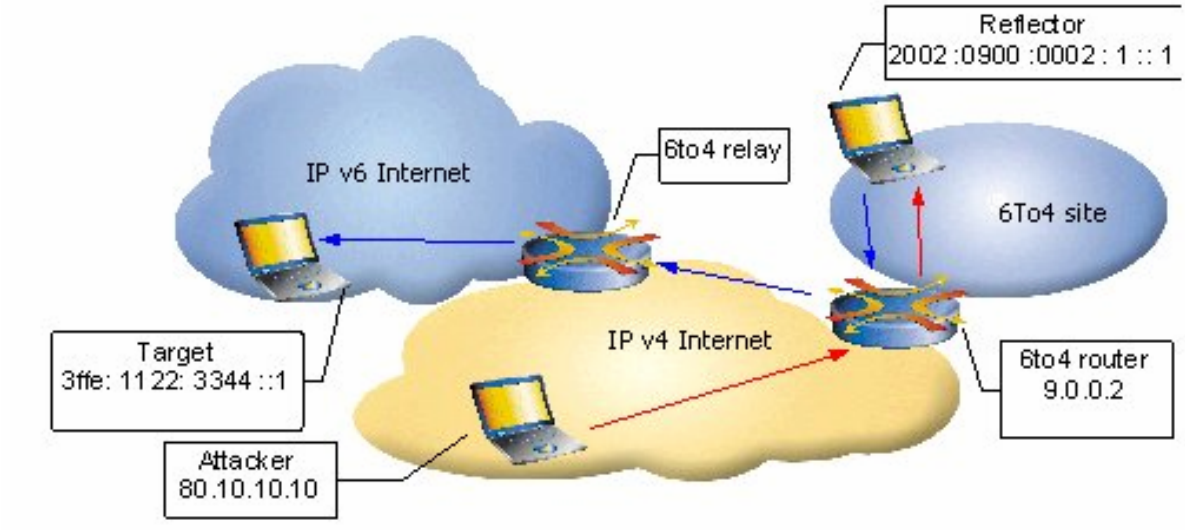

Figure 2-16: IPsec Solution

Logs Management

It is recommended that the security policy of the network operator include the filtering/auditing of incoming IPv4 packets with the IP protocol number 41 in their header, or any other encapsulated IPv6 packets.

Logs are always necessary in order to detect and demonstrate unauthorized access or another kind of intrusions. Alarms produced by IDS tools may also be useful information to be aware of security problems.

Future Work

Other important transition mechanisms exist and deserve to be studied within the Euro6IX scope, like Teredo, NAT-PT, ISATAP. The work about their security features has been initialized and is ongoing.

It is foreseen to begin with the implementation and tests of the security features that are needed in order to achieve an acceptable security level in a network using the Teredo mechanism.

2.2 Mobility and Roaming Evaluation for MIPv6 (A4.1-2)

2.2.1 AAA for IPv6 Networks

Partners

6WIND, UMU

Objectives

The main goal of this sub-activity is to provide user authentication mechanisms to IXs in order to manage users trying to access to Euro6IX network. This goal is achieved by means of:

- Use Radius protocol in IX environment in order to authenticate end-users trying to access Euro6IX network in different ways (through PPPoE protocol as it happens in DSL networks or using DHCPv6 prefix-delegation).
- Deployment of several initial applications:
  - Allow IX to provide (wired/wireless) access to IPv6 network through PPPoE connections (IPv6CP) (used in DSL environments).
  - Allow IX to delegate prefix to different networks using an AAA mechanism (for example based on RADIUS protocol).
- Allow users from a particular IX to get IPv6-network access through a foreign IX (multi-domain environments).
- Allow an easy deployment of these services by means of specific management tools.
- Replace RADIUS protocol by the more advanced AAA protocol DIAMETER and study how to make a smooth transition between them.

Description

AAA infrastructures based on RADIUS protocol are being widely used in current IPv4 networks managed by operator's domain in order to control and manage their end-users. In this way, these infrastructures will also be needed to control future IPv6 end users. Furthermore, each end user could belong to a different domain, so in order to give services to this user from a different domain, roaming must be supported. As each IX could manage end-users as a macro-ISP, deployment of infrastructures based on AAA is needed to manage these users.

In order to study how an IX could manage end-users information related with authentication, authorization and accounting and to define more specific objectives, we initially envisage several basic deployment scenarios.

<u>First Scenario (To offer support for DSL, PPP connections based on IPv6)</u>

This scenario (Figure 2-17) will be developed taking into account a unique domain. A user wants to access IPv6 networks, for example through his DSL connection with the service provider.

An end-user needs to connect IPv6 network through its local NAS (user belongs the same domain). This user is authenticated with information stored in a local Radius Server and an IPv6 prefix is obtained for this end-user from this server. So end-user could configure his IPv6 address.

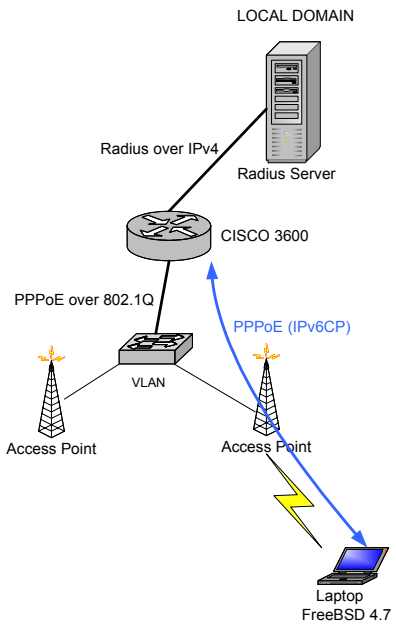

Figure 2-17: Unique Domain Radius and IPv6 UMU Test-bed

Second Scenario (Roaming)

It is based on the previous one, but now the end-user belongs to another different domain (Figure 2-18). In this case authentication information must be verified by Radius Server in end-user's domain (B). After authentication, prefix delegation must be executed.

Figure 2-18: Multi-Domain Radius and IPv6 UMU Test-bed

Another issue, which must be analyzed, is communication between Radius Servers. We envisage establishing a Virtual Private Network between Radius servers in order to improve security in communications between servers.

Third Scenario: Prefix Delegation & Radius

The following platform (Figure 2-19) is used in order to demonstrate prefix delegation over PPPoE. Integrating multiple functionalities on edge routers, PPPoE server, DHCPv6 server and Radius authentication, does this.

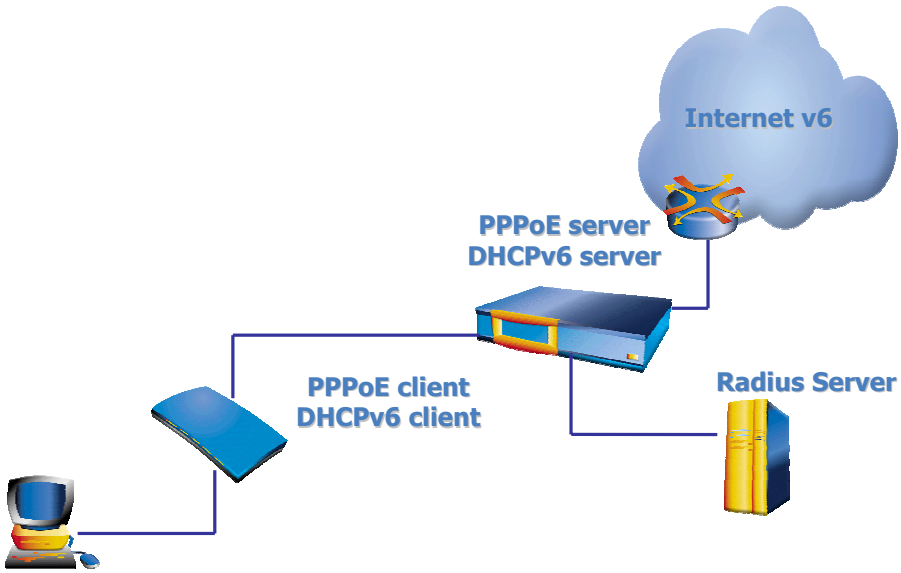

Figure 2-19: Prefix Delegation & Radius v6 Scenario

In the user area, there is an IPv6 host and a CPE (Customer Premise Equipment) that connects the user's network to the ISP network. This CPE sets up a PPPoE connection and requests an IPv6 prefix through the DHCPv6 protocol.

In the ISP network, a PE (Provider Edge) runs the PPPoE server and the DHCPv6 server, and communicates with the local radius server in order to authenticate users. The Radius server authenticates the user and the prefix is then delegated to him through DHCPv6.

Fourth Scenario: Hotspot & Radius

The goal here is to demonstrate another type of access control mechanism, which can be used in wireless hot spots. In this model (Figure 2-20), the access router applies policies and sets up automatically filters and QoS parameters. The Radius Server database contains the description of authorized customer and of their profiles.

The edge router is programmed with policies in order to dynamically react when new users appear on the access network. When new users show up, they are connected to a local web server in which they provide their identities. The server then requests the authentication by the Radius server.

By default, the edge router discards any IP packet involving a non-authorized host on the network. After the Radius Authentication, the filters are opened in order to allow the forwarding of authorized packets.

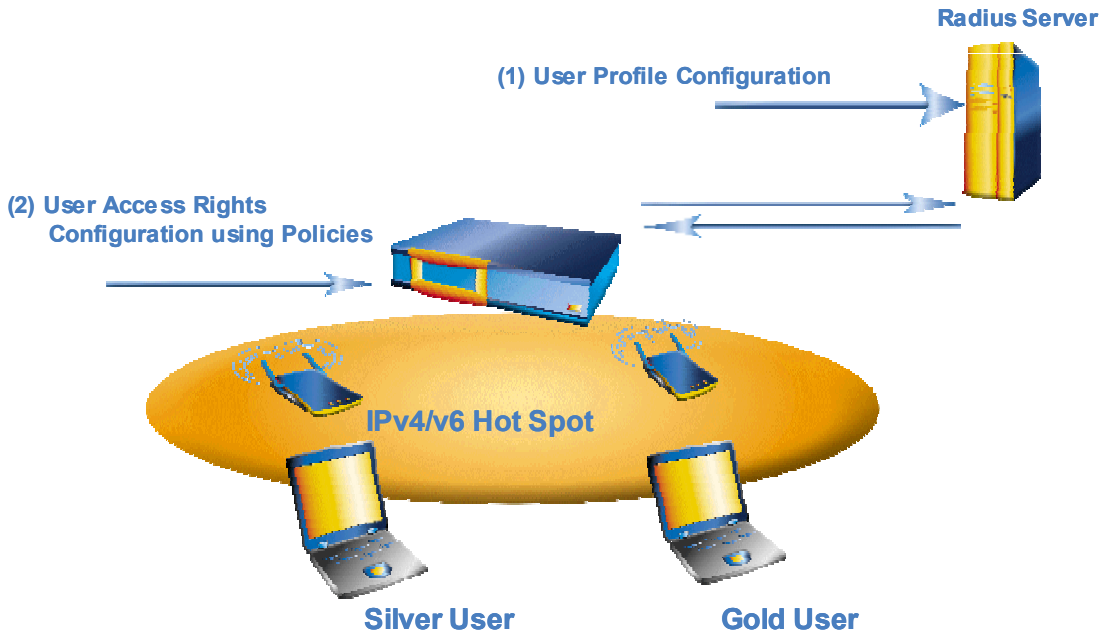

Figure 2-20: Hot Spot & Radiusv6 Scenario

Fifth scenario: Translator RADIUS-DIAMETER

The RADIUS protocol has now been obsolete by Diameter. It had many problems (i.e. security) and Diameter is a better solution for new deployment such as the Euro6IX network. DIAMETER protocol gives support to different types of network access to the Euro6IX network and therefore is better adapted to handle mobility. However, RADIUS is still widely used by TELCOS. A first step in deploying DIAMETER consists in using a RADIUS/DIAMETER translator (specified by draft-ietf-aaa-diameter-nasreq-12.txt).

UMU, based on OPEN DIAMETER (developed by sourceforge www.opendiameter.org) has developed a basic translator according to the draft and a basic DIAMETER server has been modified as well, in order to support new AVPs specified by NASREQ draft.

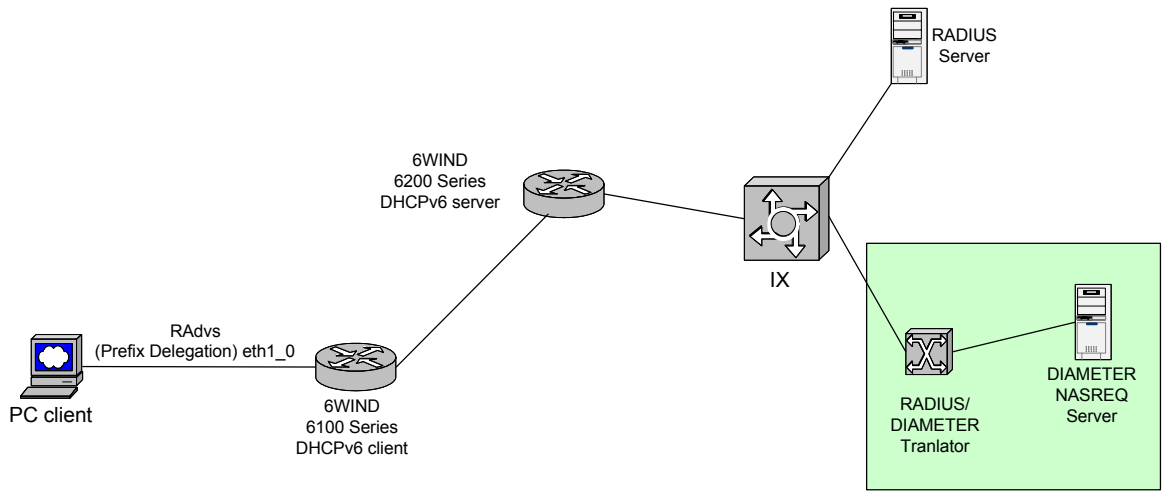

Figure 2-21: Translator RADIUS-DIAMETER Test-bed

Relevance

Access control is an important feature when deploying any commercial IP service. AAA infrastructures based on RADIUS protocol are being widely used in current IPv4 networks managed by operator's domain in order to control and manage their end-users. These infrastructures will also be needed in order to control IPv6 users.

Furthermore, a lot of IPv6 users are expected to be mobile. In fact, they will belong to different administrative domains (ISP), and roaming will have to be supported from a domain to another. Access Control will still have to be performed when arriving to a new domain and it may become an important bottleneck that could lead to service interruption during movement. So, it will be crucial to have efficient access control mechanisms in IPv6 networks.

As each IX could manage end-users as a macro-ISP, deployment of infrastructures based on AAA are needed in order to manage these users. This subject has not yet been fully studied for IPv6 networks, especially in the context of mobile users.

Results

Currently, first and second scenario have been developed and tested in UMU. So, it is possible to support PPPoE end-users roaming between different administrative domains (different IXs).

Third scenario has also been tested by UMU based on SixOS router made by 6WIND. This scenario allows prefix delegation from a DHCPv6 server, which authenticates DHCPv6 clients thanks to RADIUSv6 server. We have analyzed the procedure and we have observed which login and password of DHCPv6 client (verified by RADIUS server) must be configured in DHCPv6 server. We envisage that DHCPv6 client, instead of configured in server, should provide this login/password. The third scenario has also been tested by 6WIND.

The fourth scenario has been implemented and tested on a WiFi network by 6WIND. It worked correctly and showed the interest of having flexible access control policies in the access router. As such, these experiments only demonstrate different IPv6 access control mechanisms. The integration with Mobile IPv6 remains to be done. Finally, the fifth scenario has been tested with success in our labs besides of the Euro6IX project review in 2003 and the translator is working. It can be found in CVS in http://sourceforge.net/projects/diameter.

Furthermore, UMU has developed tools in order to manage third and fifth scenarios in easier way and allow a best deployment of these ones in Euro6IX network.

Future Work

- Design and developing new management tools for these scenarios in order to easily deploy them.
- Analyze how authentication information could be included in DHCPv6 request from client in order to avoid being included in DHCPv6 server. PANA will be considered as an option.
- Use of DIAMETER protocol instead of RADIUS protocol. Go on improving RADIUS-DIAMETER translator and adding new features based on NASREQ draft and start new scenarios based on only DIAMETER protocol (for example for authenticating PPPoE connections or even PANA connections).
- Integrate the IPv6 access control mechanisms with Mobile IPv6, and identify the problems caused. PANA protocol will be study as a possible alternative. The following figure would describe a possible experiment.

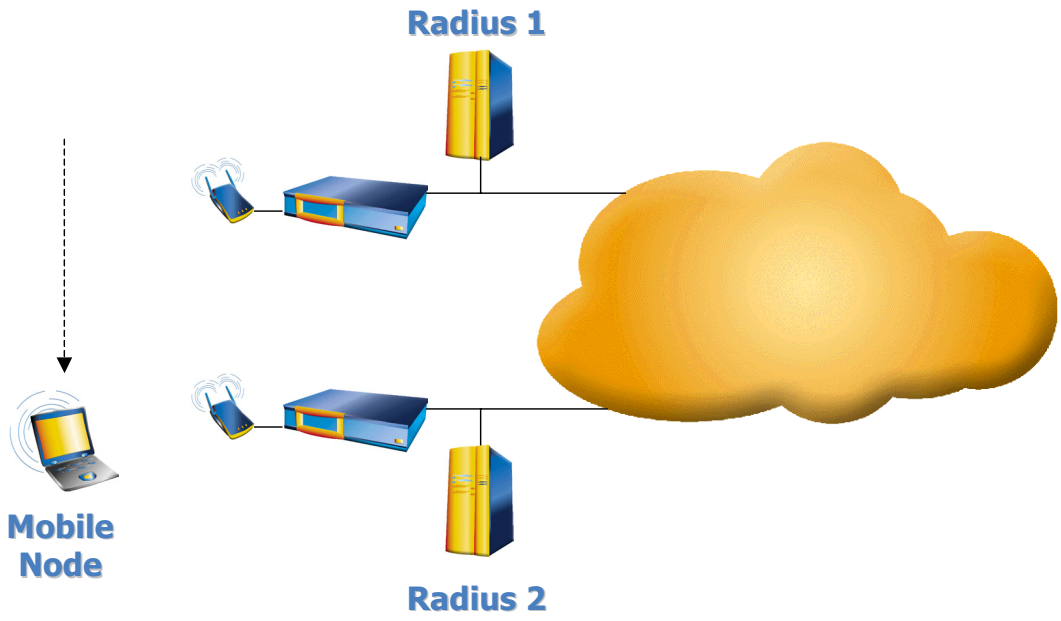

Figure 2-22: Possible Experiment Mobile IPv6 Access Control

2.2.2 UMU Test-bed Based on PDAs and MIPv6

Partners

6WIND, UMU

Objectives

- Build and test a mobility test-bed scenario based on Personal Digital Assistants (PDA) and IPv6 to allow connecting this kind of devices to Euro6IX network.
- Evaluate Mobile IPv6 implementations on PDAs.

Description

Nowadays, small devices as PDAs or mobile telephones are being widely used by end-user. One of the most important features of these devices is their mobility due to its small size and weight. An IX as macro-ISP should support and give service to these kinds of devices in order to connect them to Euro6IX network.

In order to test a Mobile IPv6 implementation for PDAs, we have used a Compaq iPAQ 3870 PDA. To evaluate a MIPv6 implementation, we have installed a Linux version for Intel ARM processors (Familiar v0.6.1). To get support for IPv6 and Mobile IPv6 it is needed to install two packages, which contain IPv6 and Mobile IPv6 modules respectively. These packages are named *IPv6-modules-2.4.18-rmk3* and *mIPv6-modules-2.4.18-rmk3*. It is also needed a package of utilities related with MIPv6 named *mIPv6-bits*.

The test-bed built and tested can be seen in the figure below.

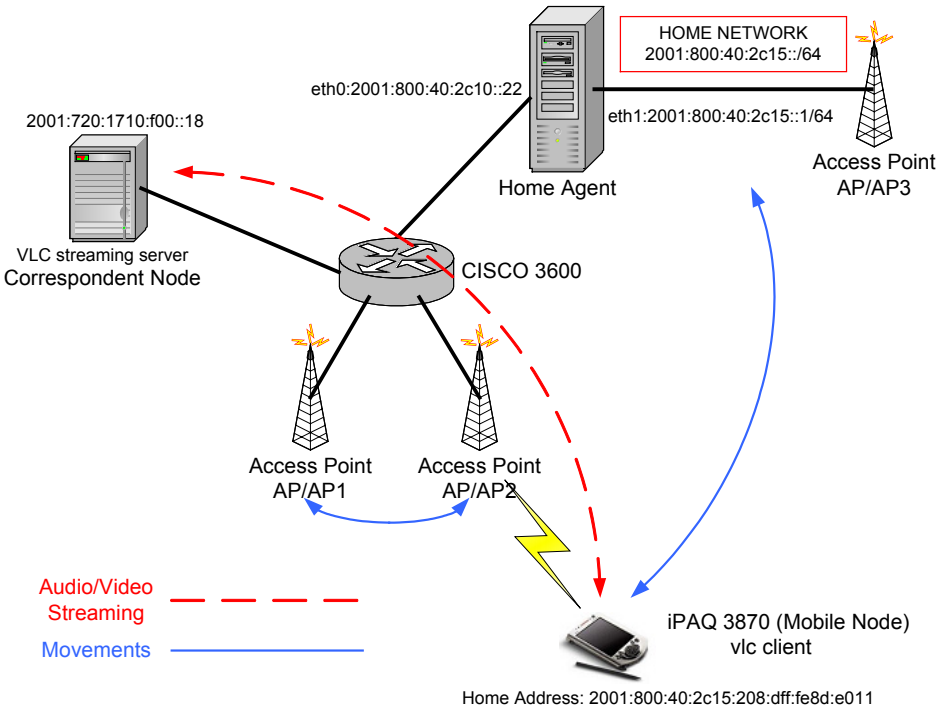

Figure 2-23: UMU Test-bed

There are three different access points and each one of them works with different frequencies. The installed audio/video-streaming server also works as correspondent node. This server sends audio/video stream to PDA's home address and thanks Mobile IPv6 protocol, this stream reach to PDA whatever network it is connected.

Relevance

Many users are using small devices as PDAs and mobile phones in the world. And it is intended to access to the network via GPRS, Wi-Fi, etc. A great feature of this kind of devices is the mobility. IPv6 provides a more optimized protocol for mobility than IPv4: Mobile IPv6. So it is very important to join both elements -IPv6 mobility and small devices- in order to give support to this kind of users in Euro6IX network. Furthermore, other technologies like UMTS are planning to use IPv6 instead of IPv4.

Results

We have carried out physical movements (PDA moves itself around the different access points coverage areas); in this case, access points placed on different rooms were configured with the same network identifier (AP). We have also simulated logical movements (PDA does not really move itself); to do this, the different access points have been configured with different network identifiers.

Normally, we have observed that the video stream stops between 2-4 seconds depending on frequency router advertisements are sent. Furthermore, thanks to this test-bed, another kind of tests with different PDAs can be carried out in the future.

Future Work

- Based on this test-bed for iPAQ and Mobile IPv6 to develop some IPv6 applications adapted for this kind of devices.
- Test other PDAs with IPv6 support on this scenario.

2.2.3 Secured MIPv6

Partners

6WIND

Objectives

The goal of this sub-activity is to implement in 6WIND routers the ability to act as a Home Agent compliant with the latest mobile IPv6 specification (currently draft 24), as well as to check its interoperability with mobile and correspondent nodes implementations.

Description

The work has consisted in adding the security functions introduced by the new MIPv6 draft in the 6WIND OS. A first step consisted in going from a draft 1 implementation to a draft 20 implementation. This was done in H1 2003 and some interop tests were made over the scenario in the following figure.

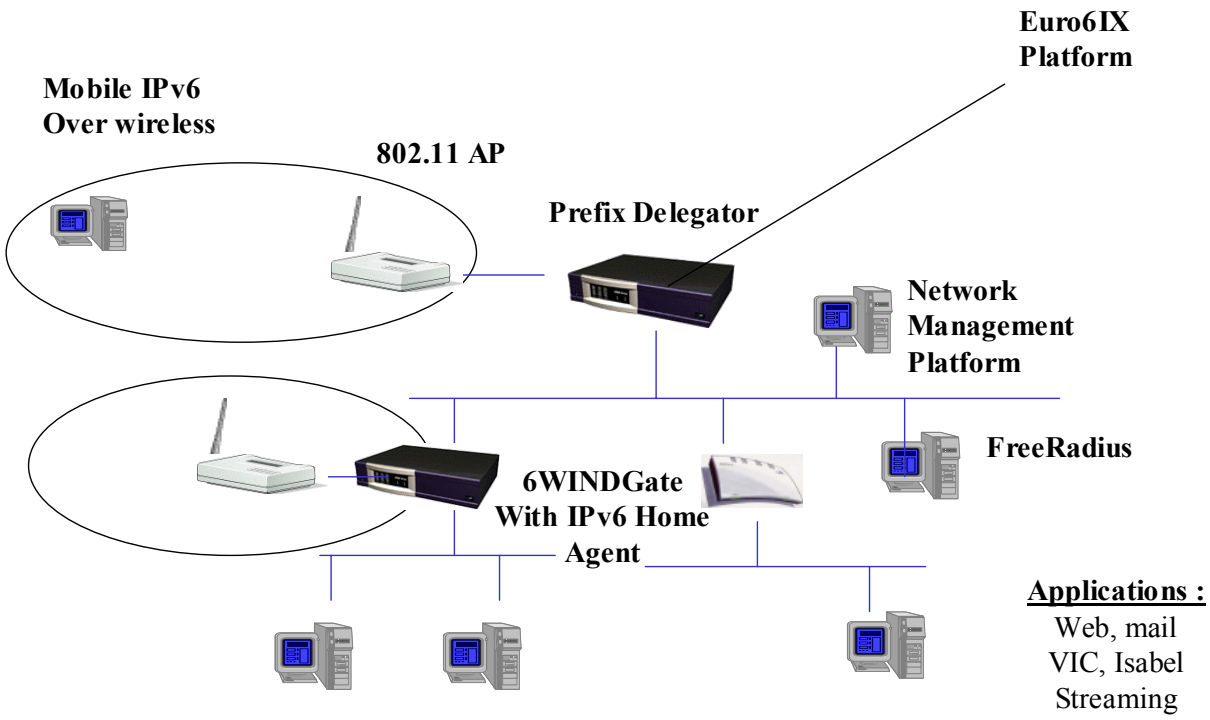

Figure 2-24: MIPv6 Scenario

The interoperability tests were carried with hosts running: FreeBSD 4.8 with KAME (Snap 21/04/03) and Linux MIPL 0.9.5 on kernel 2.4.20. Further implementation work is ongoing in order to reach the draft 24 level.

Relevance

MIPv6 is a key feature for users moving from one of the domain in the IX to another. MIPv6 must be secured, that is why implementing drafts 20 and 24 was necessary.

Future Work

Future work will consist in continuing the implementation work (draft 24 and IPsec support), as well as carrying out more interoperability tests. The integration with AAA will also be considered.

2.2.4 RADIUS and PPPv6 Integration Test-bed for RAS Service on Linux

Partners

Consulintel

Objectives

The main goal of this sub-activity is to provide user authentication mechanisms to IX in order to manage users trying to access to Euro6IX network. In particular, the sub-activity addresses the following concrete objectives:

- To allow IX to provide access to IPv6 network through PPPv6 connections.
- To provide a method for remotely accessing to the IX for management intention when a problem is detected and no other access possibilities exist.
- To provide a RADIUS based authentication method to an IX for controlling the access of telephone based users.
- To research the integration of all the needed components on an open-source platform (Linux) in order to implement a RAS.

Description

The work described here focuses on the integration of the authentication service with RADIUS and PPPv6 on Linux hosts in order to implement a Remote Access Server (RAS) through Public Switched Telephone Networks (PSTN) via modem. With this service, an operator can easily access remotely to the IX network in order to do either maintenance or troubleshooting tasks when a problem is detected and there is not other possibility for accessing to the IX network. In addition, services like stateful autoconfiguration and prefix delegation based on DHCPv6 and stateless autoconfiguration based on Router Advertisement working along with PPPv6 links are investigated in this test-bed.

The test-bed created consists of implementing a telephone based Remote Access Server (RAS) on Linux using RADIUS as authentication method and DHCPv6 and Router Advertisements for autoconfiguration purposes on the remote host.

Relevance

In order to gain IPv4 connectivity most home users use a modem with the point-to-point protocol (PPP) and they dial to the telephone number that each ISP has available for that. In this way, users use the Public Switched Telephone Network (PSTN) as an access network. Although the available bandwidth for this kind of links is very limited, they are very used (at least until now) due to both the easy for the configuration and the low cost for home users and travelers.

For offering native IPv6 connectivity to home users (without transition mechanisms like tunnels) most efforts are focused on links with more bandwidth like xDSL lines, due to the their better performance. This kind of links uses the point-to-point protocol over Ethernet (PPPoE) as link protocol in order to set up the IPv6 link. However, not all the users can afford this kind of links and they choose the legacy telephone links for getting IPv6 connectivity. In these cases the PPPv6 protocol is a must.

Furthermore, not only home users are interested in the telephone links but also professional users that want to access to a host when no more possibilities exist. Let us think, for example, in a network administrator that must remotely resolve a routing problem in an IX located at an external building when the packets cannot reach the main router in the IX due to such routing problem. In this case an IPv6 dial-in service implemented within the router can be the solution for getting access to it.

In that case it is important to have some kind of authentication method in order to avoid non-authorized users obtain connectivity to the dial-in server. RADIUS is a good candidate due to its ease of usage and the possibilities that it offers in IPv6.

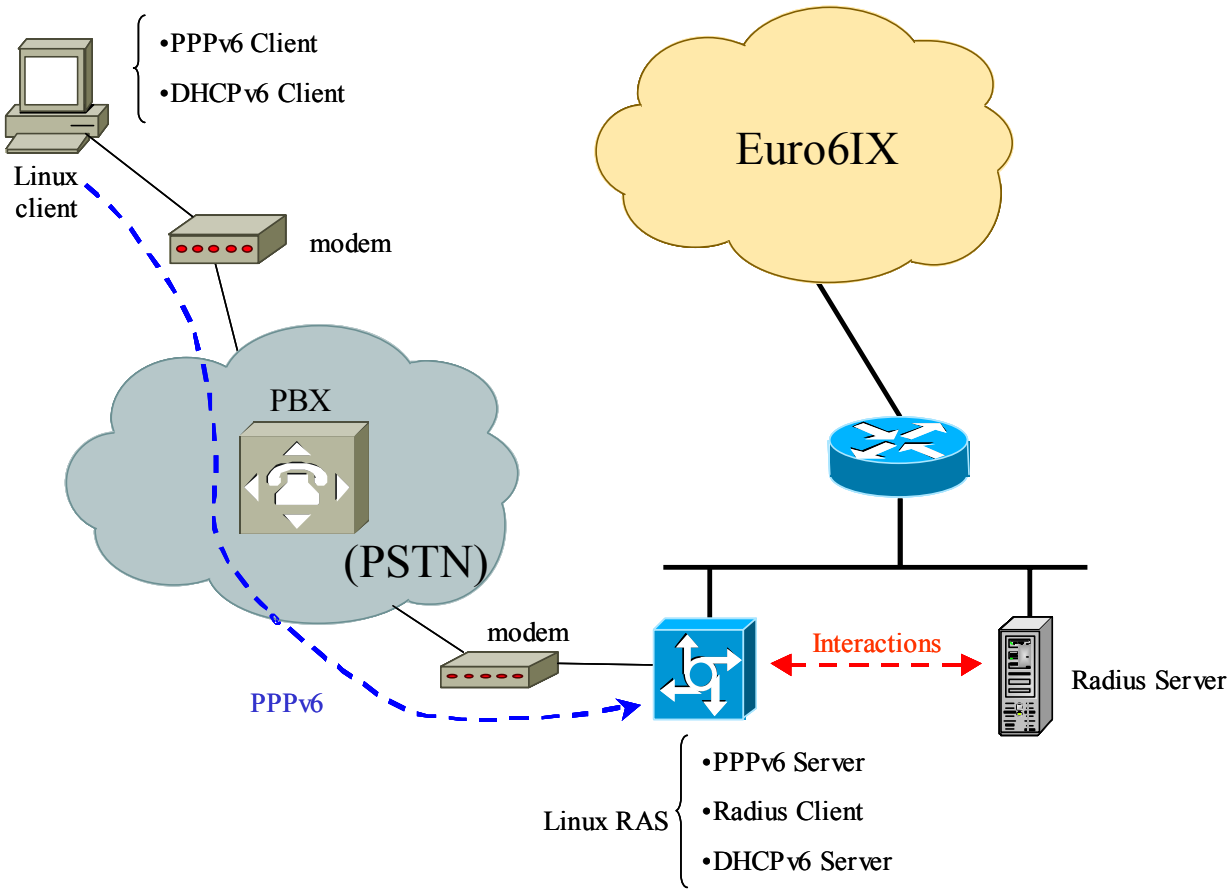

Figure 2-25: Implemented RAS Test-bed

Lastly, since the PPPv6 protocol only provides a link local IPv6 address when the link is set up other tools are required in order to provide a global IPv6 address to the remote host. Stateless autoconfiguration mechanism is a good possibility for doing that, although stateful autoconfiguration mechanism by mean of DHCPv6 seems to be more complete because it has the possibility of use the prefix delegation method and even let the remote host become a router for a LAN behind it.

All these pieces (PPPv6, AAA, DHCPv6, Router Advertisements) already exist in a separated way, but no research is yet done in a scenario where all of them are integrated on the same platform and they work joined, at least in an open-source platform like Linux. Therefore, this work tried to explore the possibilities that this scenario offers to an IX.

Results

This work shows how an AAA system like RADIUS can be used along with a link protocol like PPP for a remote user get native IPv6 connectivity without any kind of transition mechanisms.

Furthermore, different autoconfiguration mechanisms are explored in order to give a global IPv6 address to the remote host. The main conclusion of this work is that all the parts that are involved in a RAS implementation (PPPv6, RADIUS, `radvd`, DHCPv6) are available now on open platforms like LINUX and all of them can work joined in order to implement a Remote Access Server.

Future Work

Consulintel will try to implement a test bed based on the Hitachi GR2000 IPv6 router working as RADIUS client (IPv6 NAS). This work only will be possible when Hitachi provide RADIUS support to its router family.

2.2.5 TID Test-bed Based on PDAs and MIPv6

Partners

TID

Objectives

- To follow the deployment of IPv6 for mobile devices such as PDA and specially IPv6 Mobile that is to be of great interest when these mobile terminals were wide spread.
- To build a test-bed where IPv6 and Mobile IPv6 are tested over different platforms allowing access to the services provided by Euro6IX network.
- As Mobile IPv6 is not a standard yet (it is still a draft) one of the objectives is to follow its standardization status.
- To follow the availability of deployment environments for specific applications for such devices.

Description

As the mobile terminals such as personal assistants and phones are spreading and bandwidth is allowing them to use advanced applications such as streaming it is largely accepted that this kind of devices is going to become widespread. This fact and the improvement that mobility adds, will very likely speed the use of IPv6 protocol. Euro6IX network and services have to be ready to offer these devices what they require.

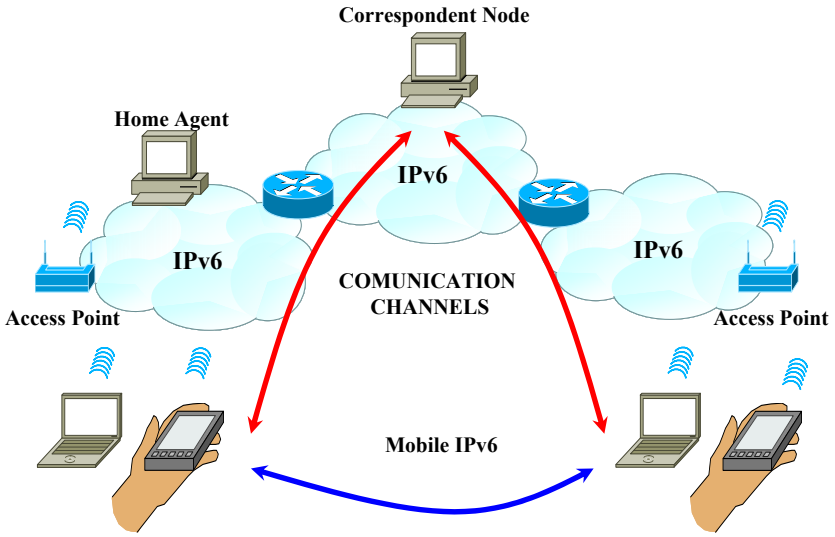

Figure 2-26: TID Test-bed

Mobility implementation by Helsinki University of Technology for laptops has been widely tested for Linux since an early draft and implementation version. It is also important to know that the only operative system for PDA that now supports mobile IPv6 is the PDA version of Linux: Familiar. Therefore and having into account that mobility implementation for Familiar is not as stable as the implementation for Linux, due to the novelty of the operative system, the test-bed has a PDA and a laptop as mobile devices and uses two Linux boxes for the correspondent node and the home agent.

Mobile implementation for PDA does not work the same for Compaq iPAQs provided with Strongarm processors than for XScale processors, i.e., iPAQs series h3800 and older have a Strongarm processor and mobility seems to work only with Familiar v0.6.1 and kernel 2.4.18-rmk3, while mobility for iPAQ series h3900 and newer seems to work with Familiar version v0.7.1 and kernel 2.4.19-rmk6-pxa1-hh13 or newer. TID test-bed has a Compaq iPAQ h3970 with XScale processor and while mobility seems to work with basic applications like ping, it still has errors with applications like video streaming.

According to the current version of the mobile draft IPv6 mobility has to allow to options: through a tunnel like IPv4 mobility (case 1) and directly using the Route Optimization mechanism (case 2). Both cases were tested with no result for advanced applications.

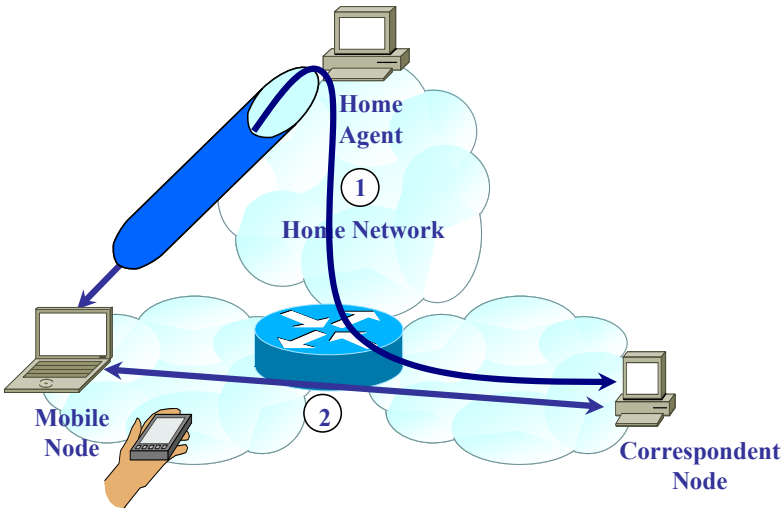

Figure 2-27: IPv6 Mobility Options

According to the status of application development environments the studies are focused in Java availability for IPv6 and compilation environments for iPAQs. Results are shown below.

IPv6 is supported in Java version 1.4 and this version is not still available for Linux nor for Familiar.

At the moment there are two compilation environments for iPAQs:

- Natively in an iPAQ cluster available for compiling. In the case of 3900 series there is an available iPAQ at ipaq1.handhelds.org.
- Cross-compiling. There are several tool-chains available at www.handhelds.org that will allow compiling software in a Linux box.

Tests have been made to compile using both methods. In the case of the native environment some applications without graphical interface have been compiled successfully and in the case of the cross-compiling environment the kernel and other free applications have been also successfully compiled.

Relevance

With the arrival of new mobile terminals and their capabilities, packet switching networks are the most likely technology to be used. In that case IP will not have enough addressing space to face that demand. Therefore Mobile IPv6 with its improvements over Mobile IPv4 is a main topic to be studied. Altogether with new access technologies such as Wi-Fi, GPRS and UMTS it represents an interesting combination for Euro6IX project.

Results

The IPv6 Mobility tests have been carried out in the following way:

- Basic and advanced services, such as ping and video streaming respectively, have been tested in the 3970 iPAQ successfully without using mobility.
- Mobility service has been run and the same services have been tried again. To change the access point to which the iPAQ was connected the "iwconfig" tool has been used. While basic services continued to work, advanced services started to give problems.

On the other hand, mobility for Linux laptops is reaching a stable point agreeing with the drafts.

Future Work

- Test other IPv6 and MIPv6 implementations such as WinCE Pocket 2003 for mobile nodes and Cisco for Home Agents.
- Access to the Euro6IX network with PDAs using other technologies such as GPRS.
- Use AAA to access the services available in Euro6IX network.

2.2.6 NPF Framework and Service API Standard for Mobile IPv6

Partners

TED

Objectives

The overall objective of the sub-activity is to design a standard framework for the implementation of the MIPv6 Home Agent functionality (MIPv6 HA for short) in an NP (network processor) technology environment. In particular, the sub-activity addresses the following concrete objectives:

- Design of MIPv6 HA NPF Software API framework.
- Specification of MIPv6 HA NPF Service API.
- Proof-of-Concept prototyping of NP based MIPv6 HA function.

Here NPF is the acronym for the Network Processing standardisation Forum.

Relevance

Mobile IPv6 in itself is interesting when researching advanced IPv6 network services because it is one of firsts IPv6 standards (if not the first) that effectively exploits the advanced IPv6 features, i.e. IPv6 destination options, routing headers and anycast addresses, which distinguish IPv6 from IPv4 (this apart from the extended address space of course).

Further, vis-à-vis the focus of the Euro6IX project on (pre-) commercial IPv6 telecommunication networks the development of an NPF software API framework for MIPv6 HA is relevant because it will lay the foundation for the appearance of commercial Mobile IPv6 Home Agent products tailored for High Performance IPv6 based Telecommunication Networks.

Finally, the work will provide an implicit validity check of the IETF Mobile IPv6 draft specification in a high performance NP based implementation environment. This could give valuable feedback to the current consolidation process of the Mobile IPv6 standards(s) undertaken by the IETF Mip6 working group.

Background

In order to meet increasing demands for higher performance, e.g. sustain Gigabit line rates, Next-generation IP Networking and Telecommunications Products are being based on network processing technologies.

In this context common API specifications for NP technology elements are believed to be essential in order to provide equipment manufacturers with the reduction in design burden and the flexibility in component choice required for low costs development and fast time-to-market.

For that purpose the Network Processing standardization Forum (NPF) has established a common software API architecture for the implementation of IP networking and Telecommunication functions based on network processing technologies and is currently in the process of adapting this common architecture framework to individual IP functions (such as IP forwarding, IP security, Diff Serv and MPLS) and specifying the associated APIs.

The NPF standards complement the “black box view” of the IETF specifications.

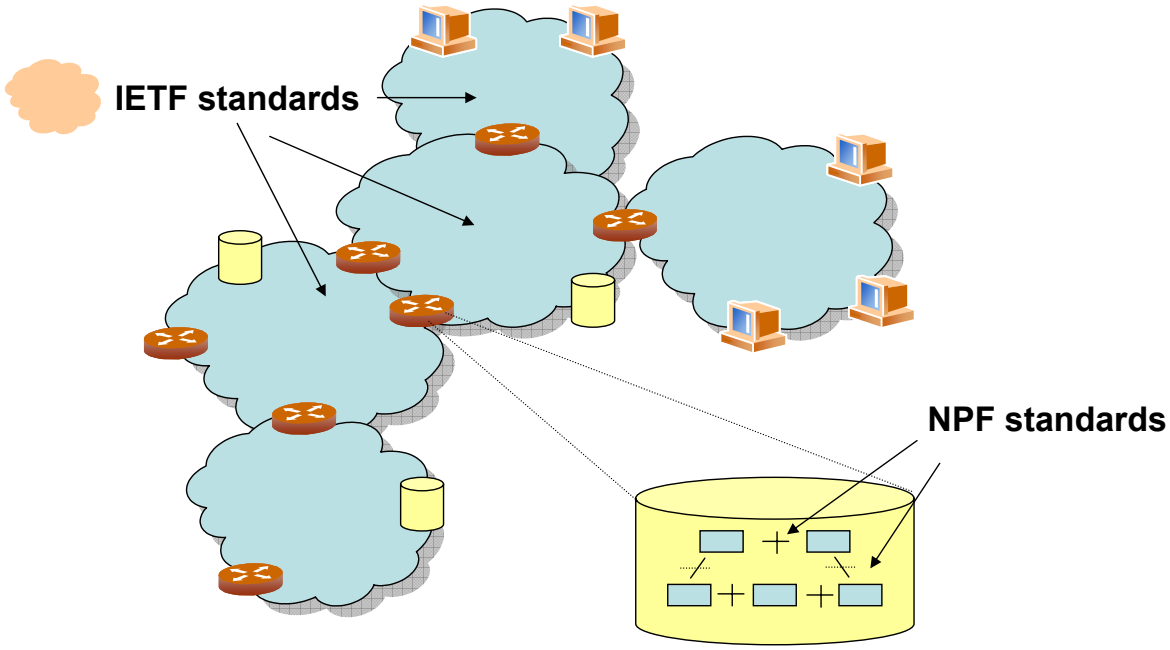

Figure 2-28: NPF and IETF Standards Relation

In the area of IPv6, and in the context of Mobile IPv6 in particular, then the following has been achieved:

- Standardization of IPv6 forwarding and IPv6 interface management was recently achieved and network processing elements supporting these specifications have started to appear.
- The framework for IP Security and the associated Service API is under ratification.

The design of a software API framework for the MIPv6 HA functionality, with its complex interaction with IPv6 Neighbor Discovery and IP Security in particular, shall provide a validation of the established IPv6 NPF standards as well as the afore mentioned validity check of the IETF Mobile IPv6 draft specification in an NP technology based implementation environment.

Description

The NPF Software API framework

The NPF framework (and technology) is based on a split of the Control and the Data Plane. The NPF Software APIs provide the means to:

- Push down configuration data from the Control Plane to the Forwarding Data Plane.
- Transfer control traffic in between the Data Plane and the Control Plane (via the NPF Packet Handler (PH) API).

The NPF Software APIs are modeled according to a three-layered architecture constituting of a data plane layer denoted the *Forwarding Data Path Layer,* an *element aware* integration and distribution layer called the *Forwarding Control Layer* and finally an *element unaware* control plane layer called the *Service Layer.*

The *Forwarding Data Path Layer* (also denoted the Hardware abstraction layer) abstracts the specific HW building blocks into Logical Function Block elements which in turn provides *Functional APIs (FAPIs)* to be manipulated by the functional element aware *Forwarding Control Layer.*

The *Forwarding Control Layer* integrates and distributes control plane configuration data to the various Logical Function Block elements (through the FAPIs). The *Forwarding Control Layer* provides element unaware *Service APIs* to be manipulated by the element unaware *Service Layer*.

The true control plane functions reside in the Service Layer. Configuration data such as, e.g., routing tables, IP Security SAs, IP Security Policies are pushed down for distribution to the appropriate elements through the respective SAPIs. The overall assumption is that each control plane function such as for example an IPv6 Routing Table Manager or an IP Security Control Module/IKE communicates with the Forwarding Control Layer through one and only one Service API. (In NPF documentations the latter is what is referred to as the Pair-wise design principle).

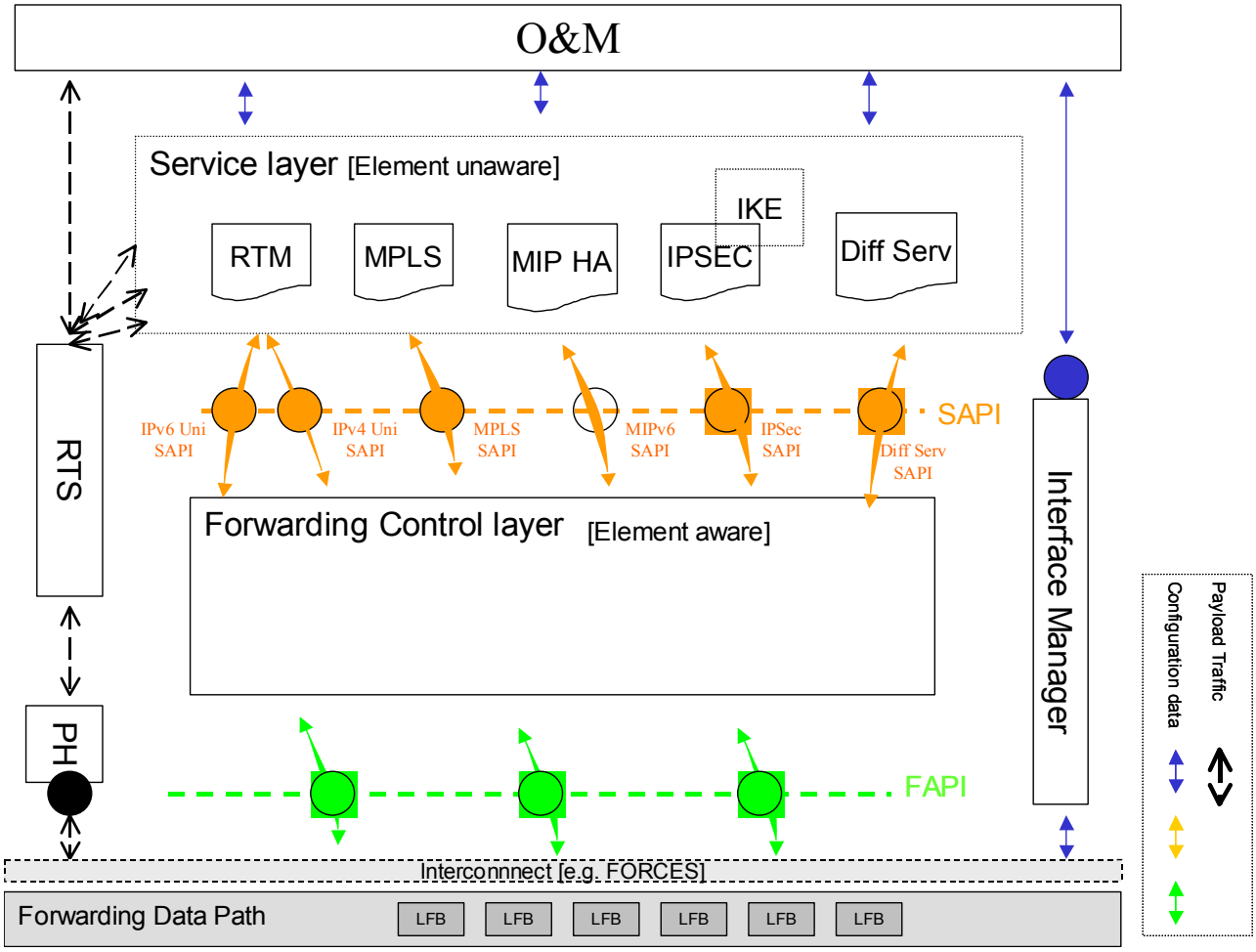

Figure 2-29: The NPF Software Architecture Framework

An NPF Software API Framework for the Mobile IPv6 HA Function

The task of defining an NPF software API framework for the Mobile IPv6 HA functionality consists in (at least) the following tasks:

- Split up of Mobile IPv6 HA functions in functions that reside in the Control Plane (slow path) and functions that reside the Data Plane (fast path).
- Definition of role of associated Service Mobile IPv6 Functional APIs.
- Definition of role of associated general purpose, as well as potential dedicated Mobile IPv6, Functional APIs.
- Specification of interaction with IPv6 Neighbour Discovery in the NPF abstraction model.
- Specification of interaction with IP Security functions in the NPF abstraction model.
- Specification of interaction with general purpose IPv6 tunnelling functionalities (if any) in the NPF abstraction model.
- Specification of interaction with IPv6 forwarding (if any) in the NPF abstraction model.

The focus will first of all be on Service API aspects and secondly on functional API aspects.

The Mobile IPv6 HA functionality scope will be the HA functions as specified in the Mobile IPv6 base specifications.

Results

The activity in Y2 of the Euro6IX project will focus on the design of a NPF software API framework for MIPv6 HA to be presented (fundamental aspects) at the NPF plenary meeting in January 2004. Additional objectives are to be addressed in Y3.

Future Work

Future work for Y3 of the Euro6IX project:

- Consolidation of MIPv6 HA NPF Software API framework.
- Specification of MIPv6 HA NPF Service API.
- Proof-of-Concept prototyping of NP based MIPv6 HA function.

2.3 QoS Deployment within IX (A4.1-3)

Partners

PTIN, TSN, BT, Consulintel.

Objectives

The QoS provisioning is a key issue for the Euro6IX network. The possibility to differentiate traffic will allow the partners and application to receive the treatment they need, contributing to the correct work of all the services.

After the isolated studies performed during Y1, the partners achieved knowledge and experience with different mechanism and functionalities (especially regarding DiffServ). At the same time, many equipment of several vendors like Cisco, Hitachi or 6WIND have been tested on different scenarios and features, which gave to the partners wide experience on their operation.

The main objective of Quality of Service activities on Y2 is to specify and deploy a QoS model to be implemented on the whole Euro6IX network. The specification of the model will include the complete set of parameters like bandwidth reservation, marking and policing rules so that the partners could inter-operate. The real deployment of this model over the Euro6IX is the final step to be achieved.

Associated to the specification of the QoS model, a study must be performed in order to collect all the information regarding the capability of the partners, taking into account that we are working in a very heterogeneous scenario and not all the partners and equipments have the same features. For that reason, a set of configuration examples for the different platforms available must be included on the Euro6IX QoS specification.

In parallel to the specification and deployment of a real and effective QoS model within the Euro6IX, it is also an intention of the project to work on other innovative QoS models. Those models are defined in order to be tested later and evaluated in different aspects. These innovative QoS models could be implemented during Y3, if they are considered appropriate for all the partners. The IntServ over DiffServ model is one of the innovative models proposed, which aims to integrate the IntServ model and the DifServ one, taking the advantages of both. Other issues are also studied like SLA agreements, DSCP/PHB coding or QoS policies. QoS propagation via BGP4+ and ECN mechanisms is already within the scope in Y2. The Flow Label applicability is already a target in the study.

Finally, a model to achieve QoS provisioning on Demand is proposed. This is a very interesting model so that the QoS can be managed easily not just by the operator but also by the final user, who will be able to select the desired QoS level at a given moment, according to their needs. This model is just a prototype and is not applied yet to the QoS specification mentioned above, nevertheless it could be a good start for this achievement.

Description

The QoS activities during the Y2 can be split in three main parts. The first one and most important, in which a QoS model is specified to the Euro6IX network, including configuration examples and collection of partners equipment and their capabilities. The second one refers to proposals of innovative QoS models or add-ons, which can improve the QoS provisioning on the network. These models could be implemented in the whole Euro6IX during Y3, if they are considered mature enough for that. The third part is related to QoS on Demand issues, where a prototype model is proposed. A similar philosophy could be used on the Euro6IX network provisioning during Y3, however many aspects must be refined for that.

QoS Specification

The QoS model proposed specifies a Premium Service besides the traditional Best Effort Service. The Premium service is intended to support real time communication, granting minimum delay and loss. It further allows throughput guarantees.

The Premium IP service to be offered by Euro6IX was created by the EU funded GÉANT project. Applying the IP Premium specification and operational guidelines produced by GÉANT seems to be an easily agreeable and proper choice for Euro6IX. The GÉANT network is equipped with routers mainly from one vendor. Euro6IX exchange points, in comparison, offer a variety of supported hardware platforms. The Euro6IX's specification proposes operational adjustments enabling Euro6IX to produce Premium IP service as originally designed.

A research aspect is introduced by supporting end system admission control and Explicit Congestion Notification. End system admission control bases on the idea that some source terminal equipment briefly sends characteristic application traffic. The receiving terminal reports the measured performance. Based on this report, the sender may mark application data for quality differentiating transport. The method's advantage is that IX routers aren't required to implement any QoS signalling protocol. Required backbone router modifications don't go beyond today's widely deployed Differentiated Services.

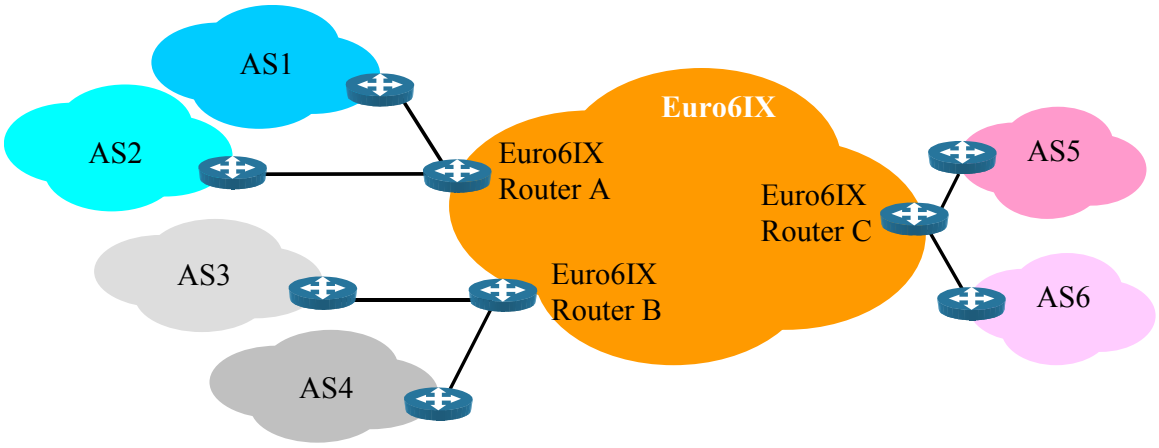

Figure 2-30: Euro6IX QoS Architecture

The main key points of this specification are the following:

- Definition of the IP Premium performance guarantees through some metrics.
- Euro6IX architecture specification.
- Destination based throughput guarantees.
- Policer settings.

- Admission Control policies.
- Monitoring.

The main key features for the involved equipment are the following:

- Classification capabilities.
- Marking capabilities.
- Policing (Dropping/Remarking) capabilities.
- Shaping capabilities.
- Queuing/Scheduling capabilities.

All these key points are defined for different behaviours, within the network of a given partner, inside the IXs and in the whole Euro6IX network.

Configuration examples are already provided for the most typical scenarios, by now, just for Cisco platform.

Innovative Models

There are some QoS models or features available that can be added to the model above, and many proposals are made in this document.

One of these proposals is the new end-to-end QoS model called IntServ over DiffServ, which tries to integrate the two QoS philosophies (IntServ and DiffServ), taking the advantages of both. With this model the operator gets the better of the two models, being able to provide strict and effective end-to-end QoS provisioning to their customers and, at the same time, in a very scalable manner. This is also a very real model because this structure of different domains is the most commonly used.

The model assumes that there are three administrative domains, both sides where customers are attached and one more, the core domain. While the former two use the RSVP/IntServ model, the later uses DiffServ. Figure 2-31 shows a schematic view.

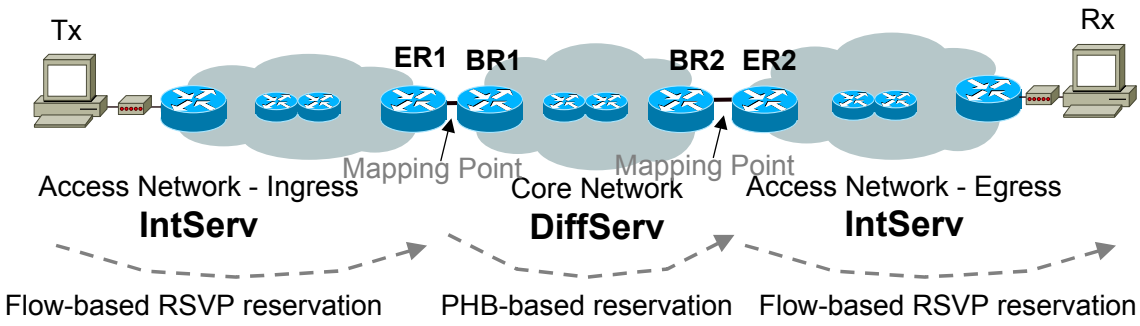

Figure 2-31: IntServ over DiffServ Model

The use of two different approaches, IntServ and DiffServ, mandates that at some point of the network the interoperation between both must happen. IntServ and DiffServ have different models for reserving resources in the network and also different approaches to lately provide the prioritization. All these integration is made on the border domains, between IntServ and DiffServ domains. So, there is the point where the mapping between the services must be performed as the Figure shows.

Besides this model, other proposals are already under study.

The advanced Control Admission model called Priority Promotion Scheme (PPS). The idea of PPS is to protect application traffic from measurement traffic while still scheduling them on the same queue. Measurement traffic is simply "tagged" and dropped as soon as a defined queue threshold is reached. In contrast, application traffic may use the complete queue before getting dropped.

The ECN (Explicit Congestion Notification) is another experimental proposal. When congestion occurs in TCP networks the normal method of dealing with this is to drop packets. This, however, is not a good choice for real-time traffic, which is sensitive to delay and/or packet loss. A solution has been proposed to use WRED (Weighted Random Early Detection) and ECN to provide routers with an alternative to dropping packets as an indication of congestion.

QPPB (QoS Policy Propagation by BGP) is another interesting proposal. QPPB allows the user to classify packets depending on access-lists, BGP community lists and AS paths. The supported classification policies include IP precedence setting and the ability to tag the packet with a QoS class identifier internal to the router.

QoS on Demand

The QoS on Demand prototype proposed can be shown in Figure 2-32. This prototype intent to let the user access to the same content available on the Internet having the capability of select the level of QoS to use. Depending of the QoS selected the content will have more or less quality (video quality on that case).

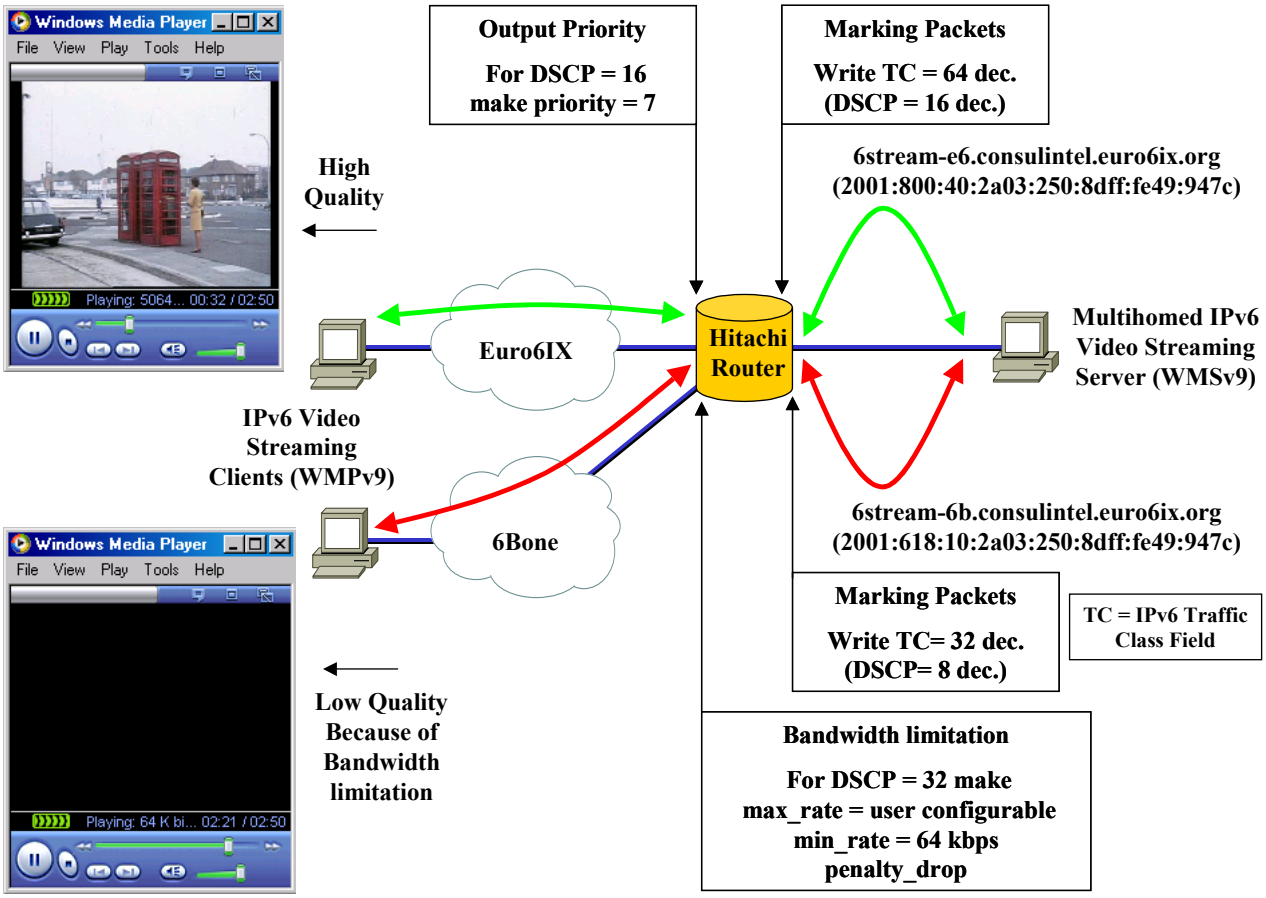

Figure 2-32: QoS on Demand Prototype

This model can be highly extended to the Euro6IX in order to control in a very easy way the QoS provisioning along the network.

Relevance

The QoS implementation is now a mandatory feature for the current networks. Since the Internet is used increasingly often for many multimedia contents this feature becomes more and more important. Furthermore, with the predicted future of the IPv6 as the network integrator, the QoS will be one of the most important issues to take care of. For that reason, since the Euro6IX is creating a pre-commercial network, this aspect cannot be disregarded at all.

Results

The specification of the main QoS model to be implemented in the Euro6IX network is finished. All the interested partners contributed and agreed with the specification, and some configuration examples were produced for Cisco platforms, exemplifying different situations. Also the information regarding the equipment and capabilities of the partners were collected, in order to know what kind of policies could be implemented.

Regarding the innovative solutions, many of them were specified like the IntServ over DiffServ model, the PPS Admission Control, the ECN with WRED congestion avoidance and the QPPB model, carrying QoS information via BGP.

Regarding the QoS on Demand, a prototype for on demand QoS configuration was specified and a practical example is available for that. This is just a prototype version allowing a multihomed machine to choose between the Euro6IX and the 6Bone path, however this model could be improved.

Future Work

As future work, the main issue to be continued is the deployment of the QoS model just specified. This requires the active participation of all the partners, analysing in particular their situation and configuring the appropriate policies, according to the specification. On the other hand, the configuration examples must be extended to all the platforms used by the partners.

Concerning the innovative QoS solutions, all of them should be further developed, in order to be tested, by at least two partners. Unfortunately, regarding the IntServ over DiffServ model it will not be possible to go further until the vendors implement this functionality. The ECN, PPS and QPPB mechanisms should be studied with more detail and check if there is any implementation available and what are the testing possibilities.

Finally, the QoS on Demand configuration must be integrated with the QoS infrastructure and extended in order to cover the configuration of marking, policing, shaping, scheduling, etc. The final goal should be the complete provisioning of the system via any API, both for the operator and the users.

2.4 Internetworking v4/v6 (A4.1-4)

The general aim of this sub-activity is to analyze the more suitable IPv4-IPv6 transition mechanisms that can be adopted inside an IX-based scenario like the Euro6IX network. In fact each IX will have to take into account the possibility to be linked to the IPv6 networks in the most transparent way, in order to make easier the transition to the IPv6 world. This work considers this scenario from an IX point of view, taking into account those mechanisms that are potentially more suitable to take advantage of the user's aggregation realized inside an Internet Exchange.

Second year's activity has been focused mainly on the study of these mechanisms and on the first testing activity of the related implementations.

Main goals to be achieved during this year were focused on:

- Tunnel Broker.
    - Theoretical analysis and testing of UMU version.
    - IPv6 over UDP over PPP mechanism for Tunnel Broker behind a NAT: Theoretical analysis and first testing of TILAB version.
    - UTMP solution for Tunnel Broker: Theoretical analysis.
    - IPv6 Tunnel Broker developed by Consulintel.
- Protocol 41 forwarding mechanism.
- NAT-PT: Theoretical analysis.

2.4.1 Tunnel Broker

Partners

TILAB, Consulintel, TID, UMU

Objectives

The purpose of this sub-activity is to analyze and to describe Tunnel Broker different solutions and implementations provided by different partners. Particularly, we have focused on the evaluation of different solutions, above all in scenarios with NATs.

Description

The Tunnel Broker model is a transition mechanism that is suitable for users/networks having dual stack devices but unable to be connected to the IPv6 networks because, for example, their ISP doesn't provide them an IPv6 native access. In this view a Tunnel Broker can be seen as virtual IPv6 ISPs, providing IPv6 connectivity to this kind of users.

The Tunnel Broker model is based on the set of functional elements depicted in Figure 2-33.

- **Tunnel Broker.** The Tunnel Broker (TB) is the intermediate entity between users and Tunnel servers. It manages tunnel creation, modification and deletion on behalf of the user. The tunnel broker can distribute the tunnels among several tunnel servers; this approach makes the TB architecture scalable respect to the number of active tunnels. The TB also registers the user IPv6 address and name in the DNS.

In order to be reached by users, that are in general connected to an IPv4 network (e.g. Internet) a Tunnel Broker must be IPv4 addressable; the IPv4 address can be a public address, if the Tunnel Broker is a public internet Tunnel Broker service, or a private address if the TB is a service offered only inside a private network. It may also be IPv6 addressable, but this is not mandatory. Communications between the broker and the servers can take place either with IPv4 or IPv6.

- **Tunnel server.** A Tunnel Server is a dual-stack (IPv4 & IPv6) router that has to be accessible via IPv4 by the users requesting the service. It is also connected to all IPv6 networks that the users will reach when they finish all the configuration steps. Upon receipt of a configuration order coming from the TB, it creates, modifies or deletes the server side of each tunnel. It may also maintain usage statistics for every active tunnel.

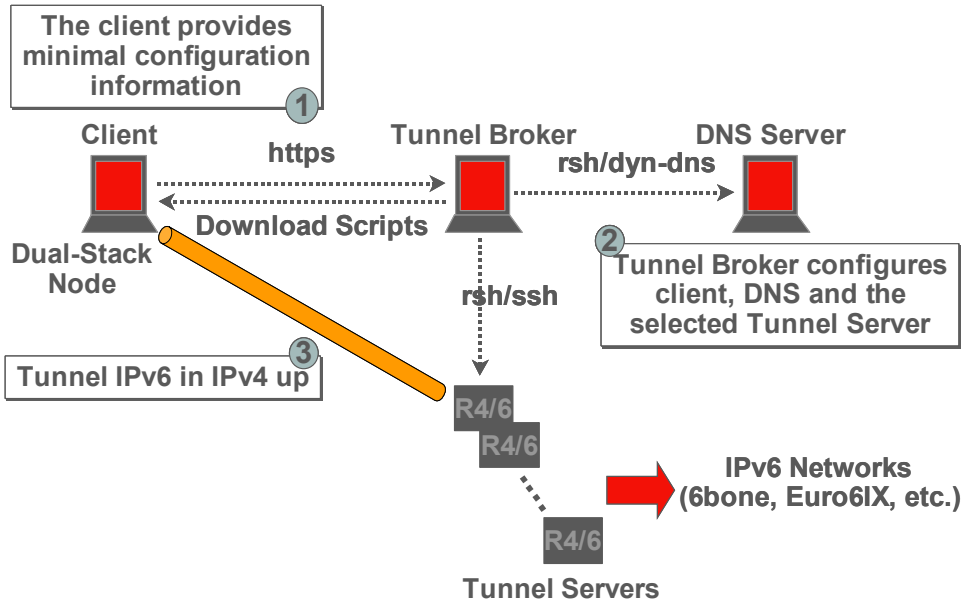

Figure 2-33: Tunnel Broker Architecture

How Tunnel Broker works

The client of the Tunnel Broker service is a dual-stack IPv6 node (host or router) connected to an IPv4 network. Approaching the TB, the client authenticates itself (e.g. using username and password) and the TB will check the credentials e.g. connecting with an authentication, authorization and (optionally) accounting server (e.g., relying on existing AAA facilities such as RADIUS). In this case the TB can be seen as an access-control server for IPv4 interconnected IPv6 users.

Once the client has been authorized to access the service, it should provide at least the following information:

- Its IPv4 address (for the client side of the tunnel).
- A name to be used for the registration in the DNS (not mandatory, it can also be stored in AAA server).
- The client function (i.e., standalone host or router).

If the client is an IPv6 router willing to provide connectivity to several IPv6 hosts, the client should be asked also to provide some information about the amount of IPv6 addresses required. This allows the TB to allocate the client an IPv6 prefix that fits its needs instead of a single IPv6 address (router tunnels).

The TB performs the following steps:

- Designates (e.g., according to some load sharing criteria defined by the TB administrator) a Tunnel Server to be used.
- Chooses the IPv6 address or prefix allocated to the client.
- Automatically registers in the DNS the global IPv6 (if a DNS name is requested).
- Configures the server side of the tunnel.
- Notifies the configuration information to the client, including tunnel parameters.

After the above configuration steps have been carried out (including the configuration of the client), the IPv6 over IPv4 tunnel between the client host/router and the selected TS is up and working, thus allowing the tunnel broker user to get access to the 6bone or any other IPv6 network the TS is connected to.

Protocols between different TB entities

All the operations described in the previous section include a sort of communication between TB and TS, TB and DNS and TB/TS and clients. The protocol used for this kind of communication is not defined and is a choice of the implementers. The main part of Tunnel Brokers use https and HTML (web pages) for the communication between client and Tunnel Broker, use rsh/ssh for the communications between TB and TS (even if some implementations could be based on SNMP) and rsh or Dynamic DNS between TB and DNS server. Furthermore there is no protocol communication between Tunnel Server and client. For this reason it is advisable to assign a lifetime to a tunnel and removed it after its expiration unless the client submits an explicit lifetime extension request.

Another solution may be to implement some kind of tunnel management protocol or keep-alive mechanism between the client and the TS (or between the client and the TB) so that each tunnel can be immediately released after the user disconnects.

There was some tentative in the IETF community to define a protocol to setup and manage tunnels configuration; the most important is the TSP (Tunnel Setup Protocol) and its evolution UTMP (Universal Tunnel Management Protocol), described later in this document.

Limitations of TB service

The main limitation of the Tunnel Broker as described in the RFC3053 is that the mechanism may not work if the user is using private IPv4 addresses behind a NAT box. This is because the TB is based on IPv4 addresses that are not translated by any box between the client and the server.

Relevance

The Tunnel Broker can be considered suitable as an Internet Exchange service because it support a big number of users and it can be considered as an high bandwidth concentration point where many users converge to access to the IPv6 networks (e.g. IPv6 internet). The Internet Exchange can offer this service to the IPv4 ISPs connected to it or directly to users that want to be connected to the IPv6 Internet via a high performances point.

The Tunnel Broker mechanisms is probably the most used transition mechanisms in the world, mainly because it is easy to install and it is suitable for ISPs to offer an experimental service and to control how many users are using the service.

Results

The main results around Tunnel Broker during the second year of the project have concentrated on the study, implementation and testing of two TB modified architectures to provide IPv6 access to "NATted" users:

- Model A: IPv6 over UDP encapsulation.
- Model B: IPv6 over PPP over UDP (NAS like approach).

The next sections describe the work done around these two proposals by UMU and TILAB, as well as the Tunnel Broker implementation activities carried out by Consulintel and the activity related to UTMP protocol carried out by UMU.

Model A: IPv6 over UDP encapsulation

The first proposal is to modify the architecture described in the RFC3053 and described in the previous section, in order to support the users behind NAT. The main approach is to redefine all the steps that are strictly related to the un-translated address approach, such as the communication of the IPv4 address by the client and the completely separated configuration between the client and the Tunnel Server.

Architecture

The following figure describe the architecture of the new tunnel broker supporting IPv6 over UDP and encapsulation behind NAT

In this case the client cannot provide configuration info because it doesn't know how its IPv4 address will be translated by the NAT. In this case after the authentication a session id will be exchanged between the client and the Tunnel Broker. The tunnel broker will configure the tunnel server sending information with this session_ID, IPv6 address to be assigned and some other parameters. The tunnel server will create a new Address/PORT UDP tunnel and will communicate these parameters to the Tunnel Broker, the tunnel broker than will send this information to the Client that will configure its IPv6 over UDP tunnel end-point using, for example, the session_ID as password.

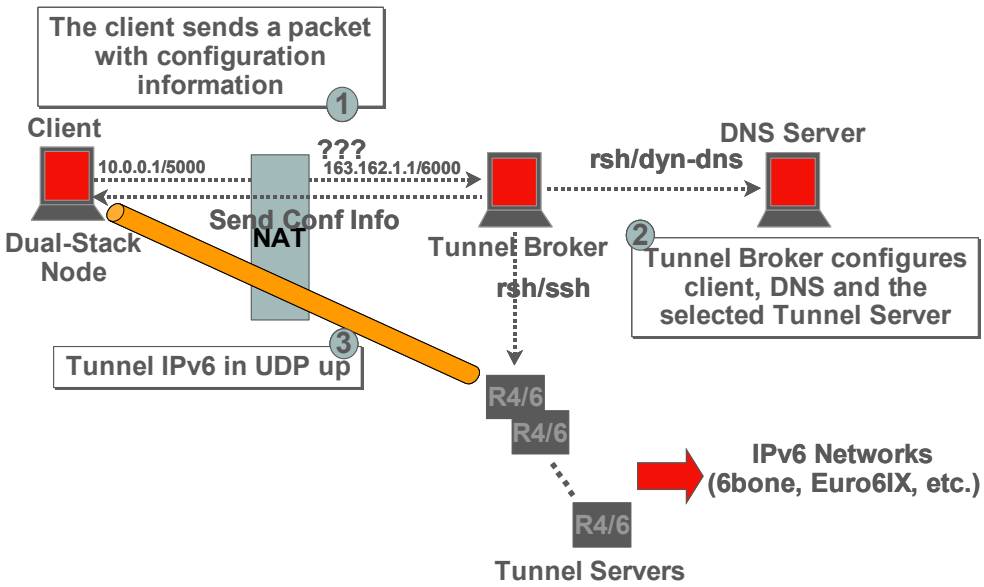

Figure 2-34: TB Behind NAT Configuration

Model B: IPv6 over PPP over UDP (NAS like approach)

Another approach could be based on PPP over UDP encapsulation that is supported by some UNIX based operating systems. In this case the approach is very similar to a RAS like approach and very different from the Tunnel Broker approach. In this case the client and the server set-up a PPP session over the UDP protocol. This PPP over UDP session follow all mechanisms described for PPP and for IPv6 over PPP IPv6CP etc.). The router/host configuration is performed in a very simple way using Router Advertisement messaging in the case of a single host, or using static configuration or prefix delegation configuration in the case of an IPv6 prefix. The following figure shows how this mechanism works.

- **Using PPP to authenticate hosts (interaction with AAA)**
- **Using RA Messages or Prefix delegation to configure hosts**

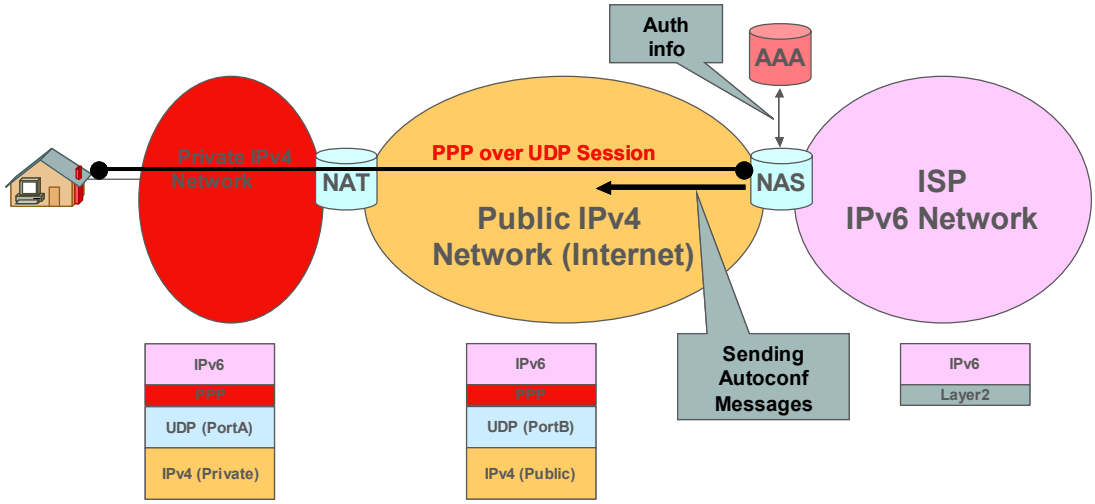

Figure 2-35: IPv6 over PPP over UDP Approach

Testing activity

In TILAB, testing activity related to Model B has been carried out using two FreeBSD machines: one is the classical client-server software and the other one is used for FreeRADIUS server.

PPP-daemon has been modified in order to allow this to communicate to the RADIUS to announce the standard RADIUS attributes. The RADIUS attribute used is Framed-IPv6-Route where a pool of addresses (normally a /48) is provided to the clients.

First experimentations gave good results, indicating how is easy for the user to get the access without too many changes.

UTMP protocol

This section describes a protocol to enable the establishment of advanced tunnels in a easy way to the end user. UTMP (Universal Tunnel Management Protocol) is based on TSP and it is thought to be used in a Tunnel Broker environment. UTMP allows to configure a tunnel like IPv6-IPv6, IPv6-IPv4, IPv4-IPv6, IPv6-UDP-IPv4 and establish the tunnel's features in a flexible way, for example, IPsec, Multicast, Routed, QoS, etc. UTMP is also a secure protocol allowing the authentication of the involved entities.

UTMP can be used in a Tunnel Broker environment or directly with a known Tunnel Server, like shows the next figure.

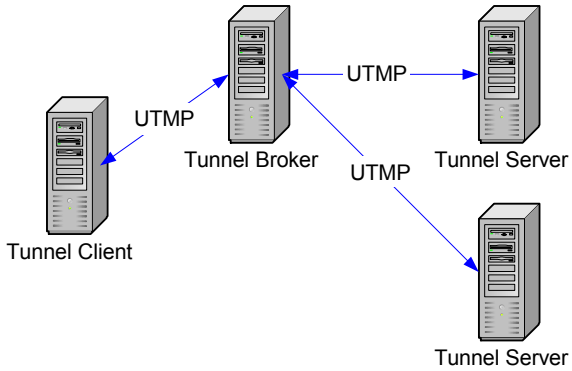

Figure 2-36: UTMP Entities

UTMP is basically based on the exchange of two types of messages between the tunnel client, broker and servers. In a example scenario, a request message allows the client to send the type of tunnel it wants to establish and the options it wants to negotiate: IPv(4/6) source, options like routing, IPsec, etc. This message is processed and modified by the broker and forwarded to a preconfigured server. The server configures the tunnel and sends a response message to the broker, which process it and send the response to the client. In the response message, Tunnel Broker sends to the client the accepted options.

The request and response messages can transport several payload options based on the client and server requirements. These options are related with the routing configuration (if the client is a router), DNS configuration (if the client wants to publish a new DNS entry in a name server), IPsec configuration (if the TunnelBroker can act like a VPN Broker), UDP tunnels (if the client wants to establish a IPv6/UDP/IPv4 tunnel), and any other future configuration needed by the client.

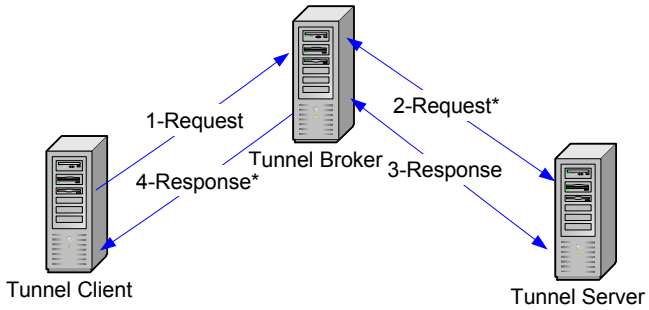

Figure 2-37: UTMP Interactions

The main advantage of UTMP is the support of any kind of tunnel and features over a tunnel using the extensions payloads. It is easy to add the routing feature or the IPsec configuration adding and interpreting the payloads in every entity and keeping intact the basic protocol. For example UTMP could be used by a VPN Broker or QoS Broker.

The UTMP protocol also allows the interactions between different Tunnel Brokers implementations, impossible in the traditional TB mechanism based on scripts sent to the clients and servers to configure tunnels.

UMU-Tunnel Broker

UMU-Tunnel Broker is a TB design to be used with the UTMP protocol. It is based on the traditional Tunnel Broker architecture.

Internetworking features. The allowed types of tunnels will be all the supported by UTMP.

Security features:

- Communication between clients and UMU-TB can be protected by:
  - UTMP with Public Key Certificate (PKC) or login/password authentication.
- Communication UMU-TB and DNS:
  - Use TSIG protocol with public keys.
- Communication UMU-TB and TS:
  - UTMP with PKC authentication.
- Integration with a VPN Broker.

Description. Figure 2-38 shows how the entities interact among them.

User enrollment. When a new user wants to be registered to use UMU-TB he has two options:

- To present a valid Public Key Certificate, validated by the UMU-TB. In the enrolment form some fields like Email, First Name, Last Name, Company or Country can be filled automatically. In this case the users only should fill the dns_name and dns_server or modify the default values. The dns_name entry allows the user to specify the IPv6 name of the final host and the dns_server allow the users to specify the name server when the entry will be created. In the next step, when the user wants to create a new tunnel he only should present his valid PKC.
- To use the login/password mechanism. In this case the user should fill all the fields in the enrolment form. The user also must specify a login/password. In the next step, when the user wants to create a new tunnel, he should present this login/password.

In both cases the new user is registered in the USER table inside the internal database.

Tunnel Request. To request the establishment of a new IPv6 tunnel, the user should be authenticated in the UMU-TB. Once authenticated, he should indicate if the tunnel would be to a router or to final host. Also the type of the tunnel should be specified.

UMU-TB selects a new Tunnel Server from the TS database and selects a new IPv6 address (or prefix) for the client, the new tunnel is added to the TUNNEL table.

The new host in the name server specified by the client or in the default server should create a new DNS entry. This entry will be created using DNSSEC (TSIG protocol), and the DNS server specified by the user should be previously registered in the DNS table. If there is no a DNS server specified by the user, the default DNS server should be used. In both cases a DNS name entry should appear for the new TB client.

To establish the IPv6 tunnel, UMU-TB configures the Tunnel Server selected by means the UTMP protocol indicating the needed information to create the new tunnel.

The last step is to configure the Tunnel into the client host. The UTMP response message contains the client features to be configured in the end device.

IPsec tunnels. If a user wants to establish a IPv6/IPsec over IPv4 tunnel he should notify it in the tunnel request step. The user can select IPsec options in the request form or to use the default options. When a new IPv6/IPsec over IPv4 tunnel is created, UMU-TB adds a new entry to the IPsec table linked with the TUNNEL table, this new entry store the IPsec configuration of the tunnel. The IPsec modes allowed in the configuration should be AH, ESP or both headers in transport mode.

In this case, both the client host and the Tunnel Server should have an IPv6/IPsec stack.

In an advanced scenario UMU-TB could be integrated with a VPN Broker to configure the IPsec Tunnels when the client of the tunnel is a only IPv6/IPsec host/router.

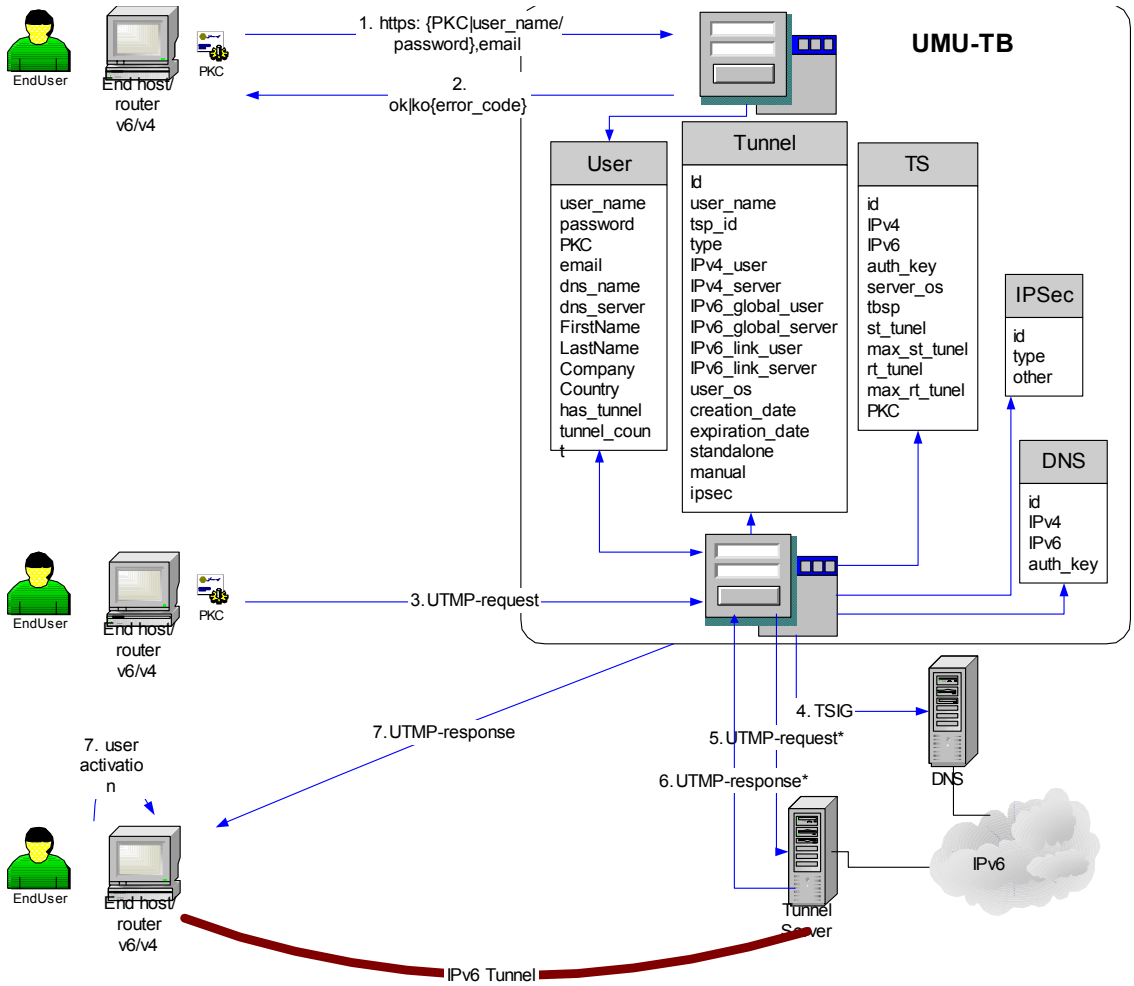

Figure 2-38: First UMU-TB Design

Tunnel Broker experience in Consulintel

The Figure 2-39 shows the reference scenario used by Consulintel for its activity regarding Tunnel Broker.

A Web Client/dual stack PC connects with tunnel broker at http://tb.consulintel.euro6ix.org, that is a LINUX server that will serve the pages to the client for requesting an IPv6 tunnel between this PC and the IPv6 capable router.

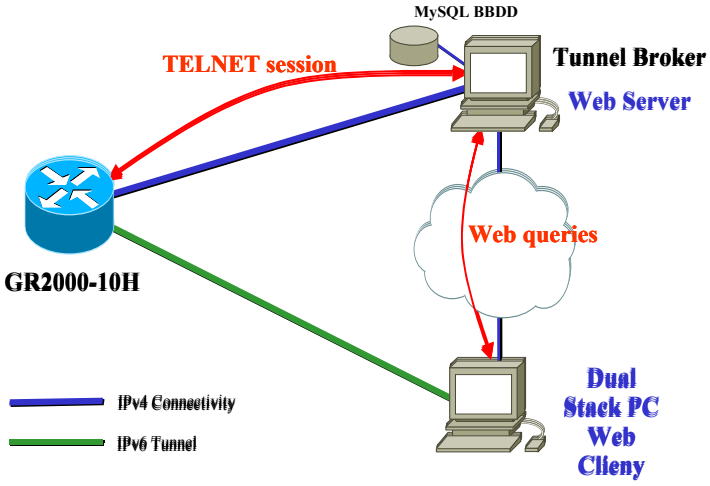

Figure 2-39: Tunnel Broker Structure

Next, the tunnel broker establishes a telnet session with the router and configures the tunnel in the router-side, and shows in the client PC the script that should be executed in it to configure the tunnel in the client-side.

The information is stored in a database, so it will be always a register of data of users and tunnels.

Next figure shows the general aspect of the web tool implemented for the Tunnel Broker.

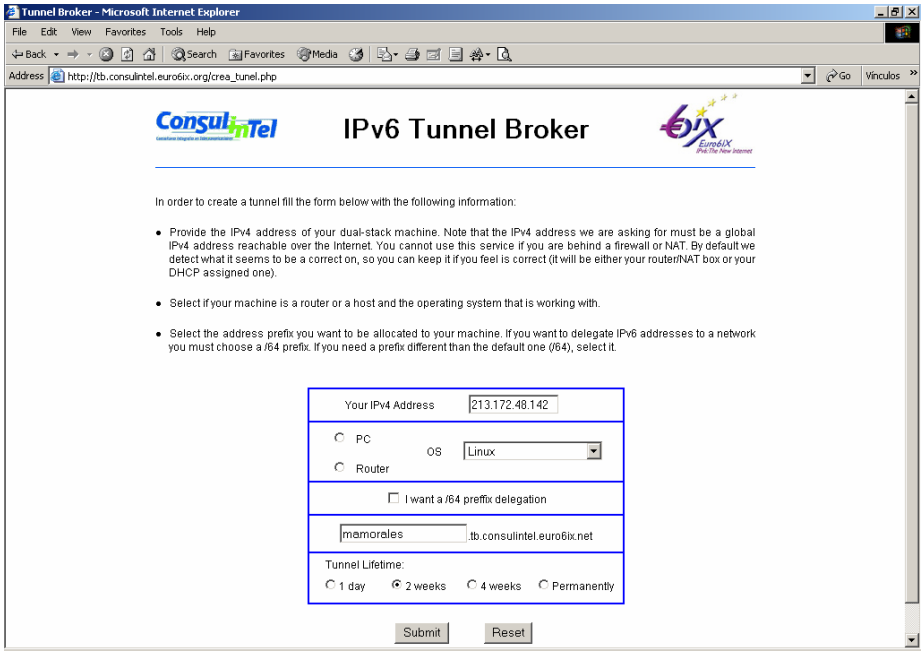

Figure 2-40: Tunnel Request Page

Future Work

Since the importance that the Tunnel Broker has to speed-up the transition between the IPv4 and IPv6 world, it will be helpful that this activity continue with the aim to optimize the results obtained, carrying on the testing activity related to the protocols and mechanisms not still fully tested. For example, it could be interesting investigate on the security implications of UTMP (since UTMP ensures the authentication of the entities but not the ciphered, using SSL or IPsec between brokers and servers).

More activity in this direction will be carried out by Consulintel, which will improve these services and adding new functionalities to make this tool as useful as possible. One of these new functionalities is the tunnel creation by sending ICMP ping packets of a given size to the LINUX server that works as the Tunnel Broker. That it will be very useful for those users that do not have the possibility of connect to the tunnel broker by means of HTTP protocol.

Important aspects to consider are moreover the usage scenarios and where these solutions could be interesting and the possible interaction and complement to actual IETF initiatives.

2.4.2 Forwarding Protocol 41 in NAT Boxes

Partners

Consulintel, UPM

Objectives

The general objective of this sub-activity is to study and document the mechanism that allows the direct establishment of IPv6 tunnels over IPv4 through NAT boxes. In particular, the sub-activity addresses the following concrete objectives:

- Study and test proto 41 forwarding functionality in detail and its support by the different NAT boxes manufacturers.
- Document that functionality and progress it as an internet-draft inside v6ops IETG WG.

Description

Most of the existing solutions for the transition to IPv6 rely in tunnels assuming that the client end-point is an IPv6 capable router. However, nowadays the installed base of IPv4-only NAT boxes/routers is still quite big, while most of the client operating systems already support IPv6.

The ability of some IPv4-only NAT boxes/routers to establish IPv6 tunnels from systems inside the private LAN (even using private IPv4 addresses) to routers or tunnel servers in the public Internet has been used for some time. However, it has not been documented so far.

The basic scenario of the mechanism presented is shown in next figure As it can be seen, a Tunnel Client (a host or a router), which is connected to Internet through an IPv4-only NAT box using a private IPv4 address, establishes an IPv6 tunnel to a Tunnel Server with the help of a Tunnel Broker. The mechanism can also be used without a tunnel broker, ending the tunnel in an IPv6 router, which is configured manually.

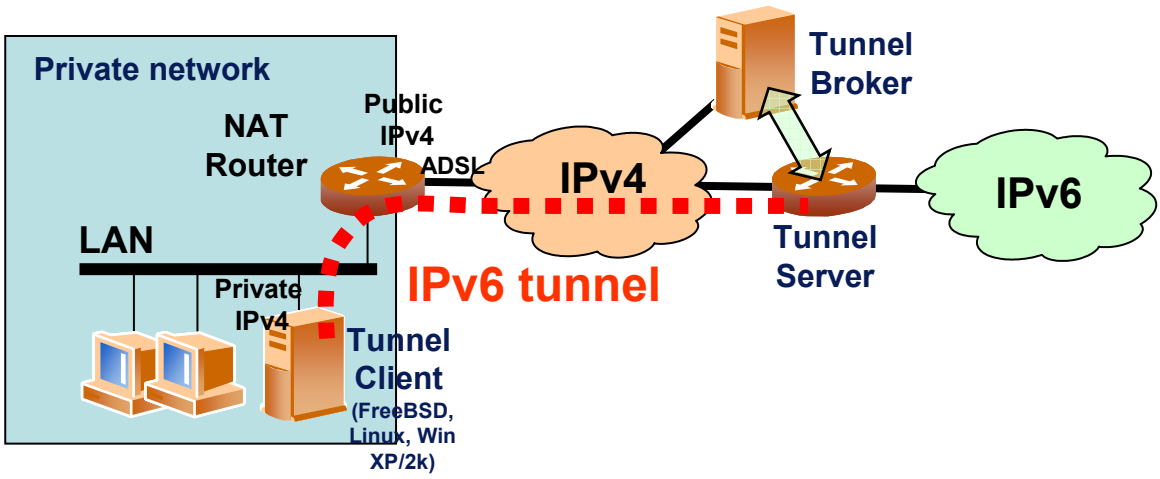

Figure 2-41: Basic Proto-41 Forwarding Scenario

Relevance

Exploring and documenting proto 41 forwarding mechanism can be consider as an important issue for IPv6, because it can help in the deployment of IPv6 over NAT based SOHO scenarios. Although it is not a transmission mechanism, it can be used as a temporary fallback solution to provide IPv6 on those scenarios when neither native IPv6 nor 6to4 mechanisms are available.

This mechanism is not usable on all existing IPv4-only NAT boxes/routers. However, the large number of them that already support it gives an opportunity to rapidly deploy a huge number of IPv6 nodes and networks.

Results

The functionality has been successfully tested over several NAT boxes manufacturers and several users to connect to Euro6IX network are currently using it.

Three drafts of the document describing proto 41 forwarding functionality (draft-palet-v6ops-proto41-nat-0x) have been submitted to v6ops IETF Working Group and it has been presented in 57th and 58th IETF meetings. A request for accepting it as a WG item has been recently submitted and it is being discussed.

Future Work

This task will continue during Y3, completing the work already done in terms of tests over different NAT implementations, completing the manufacturers survey and progressing the document in IETF.

2.4.3 NAT-PT

Partners

TILAB, UMU, FTRD, TID

Objectives

The purpose of this sub-activity is to analyze and describe NAT-PT like an IPv4/IPv6 transition mechanism that could be in principle considered as a Layer3 IPv6 Internet Exchange service.

Description

NAT-PT (Network Address Translation-Protocol Translation) is a transition mechanism that belongs to the translation mechanism group. The objective of NAT-PT is to provide IPv4 users or networks access to IPv6 and the other way around.

NAT-PT as an Access Service

NAT-PT could be one of appropriate transition mechanism for a status such as the current one: There are IPv6 networks with IPv6 services but there are still a lot of IPv4 services. When there are IPv6 networks, one of the objectives of transition is to allow those IPv6 users to access to the already available IPv4 services. This states the first stage of IPv4 to IPv6 transition. As those IPv6 networks are deployed and begin to evolve, developing IPv6 services, IPv4 users may be interested in accessing to those services. This states the second stage of transition. Finally the situation is IPv6 users accessing IPv4 services and IPv4 users accessing IPv6 services. In other words, IPv4 and IPv6 networks connected.

In summary, three steps have been defined related to NAT-PT:

1. IPv6 users to IPv4 networks.
2. IPv4 users accessing IPv6 networks.
3. IPv4 networks to IPv6.

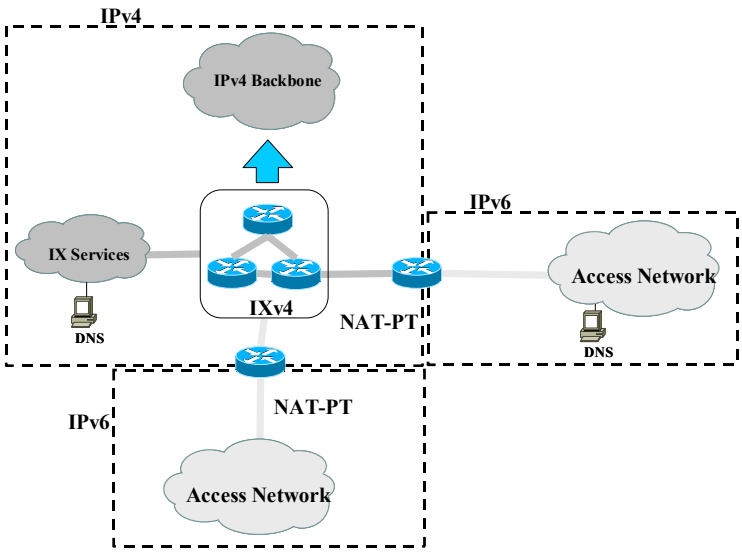

Figure 2-42: NAT-PT – IXv4

First two steps are related with the access of the users to the network. This means that the NAT-PT feature should be considered an access service and thus it will be better provided by ISPs. This is the case of the current Internet. If an IPv6 network wants to get access to the IPv4 Internet the scenario is shown below (Figure 2-42).

The next scenario (Figure 2-43) shows the situation where IPv4 users want to access to IPv6 services through an IPv6 core network. In this case the IPv4 addresses are translated before getting the IX. The IX do not have to work with IPv4 address, i.e., it is an IXv6 only. This would be the case if IPv4 network wanted to connect to the Euro6IX network.

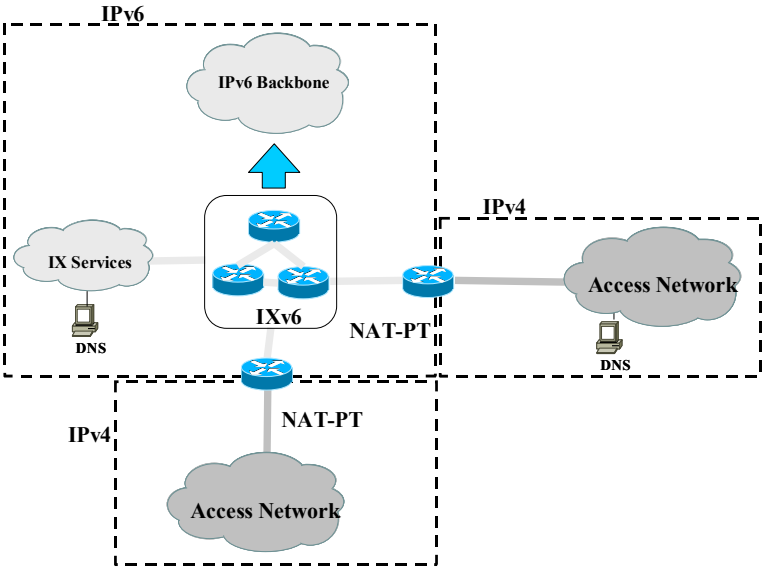

Figure 2-43: NAT-PT – IXv6

NAT-PT as an IX Service

Once that IPv4 core networks and IPv6 core networks are deployed the question of how to connect them arises. There are several options; two of them are show here.

1. To have NAT-PTs between IXs.
2. To connect the access networks to the IPv6 IX.

The first option is shown in Figure 2-44. This is the case where the exchanges are connected through NAT-PT boxes. This would be a critical point, where NAT-PTs need to have a high processing rate and stability. Both reasons make NAT-PT unlikely to be put between exchanges because it has to be understood as a transition mechanism and exchanges are thought to be as much stable as possible with the least changes to be made.

Another option is to have Dual Stack exchanges so that any access network could be directly connected to the IX. As before, exchanges require high processing rate and stability. This situation could be simplified by bringing the NAT-PT boxes outside the IX, as shows below (Figure 2-45).

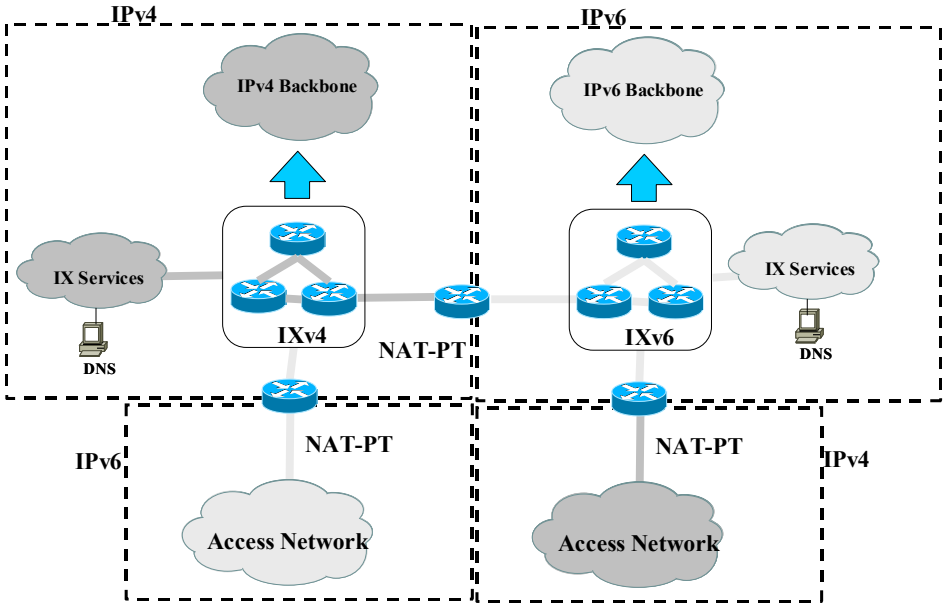

Figure 2-44: NAT-PT – IXv4 – IXv6

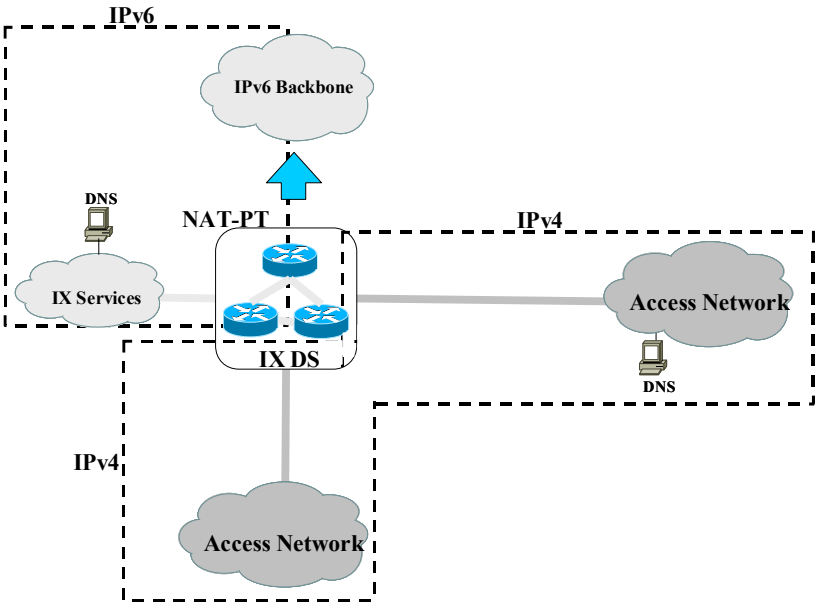

Figure 2-45: NAT-PT – IX Dual Stack

Relevance and Results

NAT-PT pros and cons

The goal of this section is to list the pros and cons of the translation mechanism NAT-PT and to evaluate the impact and the relevance of this mechanism within an Internet exchange as service.

Pros

The transition mechanism NAT-PT aims at enabling communication between IPv4-only nodes or applications with IPv6-only nodes or applications and vice versa. NAT-PT is based on SIIT (stateless IP/IMCP translation algorithm) that describes how to translate an IPv4 header into an IPv6 header and vice versa.

However SIIT, as an algorithm, does not provide any clue about addresses correspondence from a version to another. NAT-PT includes, then, the features necessary to enable translation of the address fields from a version to another. A mapping between IPv4 and IPv6 addresses is performed on a dynamic basis and is associated with a temporary context to a given communication.

- *Architecture.* From an architectural point of view, the use a NAT-PT is simple since it just needs to install between an IPv4 network and an IPv6 network, providing then some means of network interconnection.
- *Impact on end points.* NAT-PT is installed at the border of and IPv6 network and an IPv4 network, and does not need any modification of any kind of the nodes or applications. As a result, an IPv6-only node will see any IPv4-only nodes as a IPv6 node and vice versa.

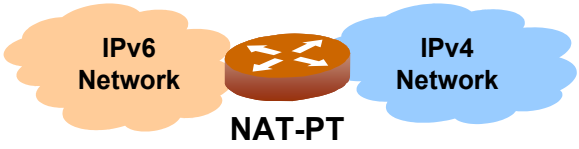

Figure 2-46: NAT-PT Network

Cons

- *DNS-ALG.* Besides SIIT, NAT-PT is also made up of a DNS-ALG (Application Layer Gateway) that replaces any single A record by an AAAA record (and that collaborates to the temporary context set up, as well). Such a change is necessary to enable an IPv6-only application to communicate with an IPv4-only application, since the DNS resolution of the destination must produce a result that fit with the single IP version the source is using.

  This behavior results in a strong constraint: Any DNS traffic MUST go through the device that runs the DNS-ALG.
- *Stateful mechanism.* Since the mapping between IPv4 and IPv6 addresses is associated to the context of a given communication, all traffic of this communication MUST be routed to the same unique NAT-PT box. Any fault of the NAT-PT will result in a communication failure, unless the implementation is able to recover broken context.
- *Header translation.* SIIT algorithm has a number of limitations, since, for example, a number of fields of the IPv4 header have been suppressed in the IPv6 header, or some IPv6 ICMP messages have no equivalent in IPv4.

  Furthermore, SIIT applies only to IP header. As a result, any information contained into IPv6 extension (e.g. mobile IPv6, authentication or ESP extension header) will be lost through out the NAT-PT mechanism. IPv6 applications using extension header will need additional artifacts to pass through NAT-PT tool.
- *Application embedding IP addresses.* This kind limitation is inherent to address translation mechanisms. Some applications are embedding IP address into IP payload. So, an application dedicated ALG must be associated to the address translation mechanism, to enable the concerned application to properly work across the translation mechanism.

  NAT-PT is part of this translation mechanism, and has the same constraint. Moreover, like the well-known IPv4 NAT-PAT mechanism, and for the same reasons, IPsec based application cannot pass through NAT-PT.

Architecture impact

One main architectural constraint related to NAT-PT comes from the DNS-ALG that must process all DNS traffic.

Today the DNS root is only available over IPv4. Then, a name resolution done over IPv6 transport would have to be transported over IPv4, if no IPv6-accessible cache server is able to answer the query. This condition must be met when the name resolution applies to an IPv4-only domain name. Hence, the DNS-ALG must be the bridge enabling the DNS query to move from IPv6 transport to IPv4 transport, and vice versa.

Since this rule must apply to any IX client in any case, the DNS ALG must be located into the NAT-PT box, according to the drawing below. In addition, the DNS-ALG must be part of the resolution chain of any given name.

Relevance as IX service

Usually opinions differ on whether or not NAT-PT is relevant of an ISP service or a backbone service. The same kind of debate may rise on whether or not NAT-PT is relevant of an IX service.

The primary goal of an IX is to enable entities operating large networks to exchange routes to each other, in order to gain larger and better connectivity, namely at the IP layer. A facility providing network interconnection, i.e. between IPv4 an IPv6 network is indeed a relevant of an IX, since it provides the same kind of benefits, at the same layer.

Factually, NAT-PT provides some means of network interconnection, but on statefull basis, and based on the use of DNS-ALG that acts at the application layer. NAT-PT appears as a rather sophisticated stateful mechanism made up of IP layer and application layer components, which have something in common with the well-known IPv4 NAT-PAT mechanism.

Besides, the assumption according to which, pure IPv6-only network may exist seems to be bogus, since any name resolution out of the given pure IPv6 network domain name, will need the help of IPv4 transport for the DNS queries, due to the lack of name root IPv6 reachable.

Suitability of the NAT-PT in an Internet Exchange

It is important not to loose the goal of NAT-PT. It is a transition mechanism and that means that its aim is to facilitate the transition to IPv6 and disappear after the transition is complete. Since the beginning of IPv6 exchanges deployment, they have to be the most possible stable.

Current NAT-PT implementations work at application level. This implies a slow translation rate, putting a bottleneck in the path of any packet going through it. Having in mind that NAT-PT is a single point of failure, considerations about redundancy and load balance should have to be taken into account.

Future Work

Although NAT-PT is an interesting option in the context of v4/v6 interworking, several issues are still to be further evaluated. One of them is the performance, scalability and throughput of NAT-PT in the context of a widely deployed service, especially when all DNS traffic has to go through NAT-PT. In that sense a work to be considered in the next months is the evaluation of the interaction between NAT-PT and DNS, and search solutions to use DNS in IPv6 isolated networks.

2.5 WP2/WP4 Common Evaluation Activity (A4.1-5)

The main objectives of the WP2/WP4 Common Evaluation sub-activity are:

- The evaluation of topics that implied preliminary decision or evaluation from WP2 or deployment by WP3.
- Theoretical activity that will promote in some moment the launching of new activities.

During Y2, mainly two areas of work have been active inside A4.1-5 sub-activity: Multicast and Multihoming. The current status of both is described in the following sections.

2.5.1 Multicast

Partners

Consulintel, 6WIND, BT, TID, TILAB, TSN, UMU, UoS, UPM

Objectives

Multicast common evaluation objectives include:

- Infrastructure
    - Deployment of Multicast RP in the 6IXs.
    - Study, test and deployment of Interdomain IPv6 Multicast (IX-Customers, Euro6IX-6NET, etc.).
- Applications
    - Provision of enabled Multicast services as VoD, Music/Video Streaming, HDTV.
    - Provision of authentication for Multicast services.
    - Study of new Multicast applications (development/porting).

Description

During first half part of Y2, Euro6IX network has been connected to the M6Bone network through several partners as the Figure 2-47 shows, but this is a transition scenario until the Euro6IX IXs will be connected directly to M6Bone network.

From a proposal led by UMU and TILAB, done within WP3, there were two possible scenarios for Euro6IX to gain M6Bone connectivity.

First scenario

Euro6IX network uses the connections already up between its partners and the M6Bone network to interconnect to the M6Bone RP managed by Renater. This solution seems to be easier to realize than the second one described below and could be adopted in a first phase to make sure the interconnection in the short time among the two networks and to provide the multicast service to the IX users.

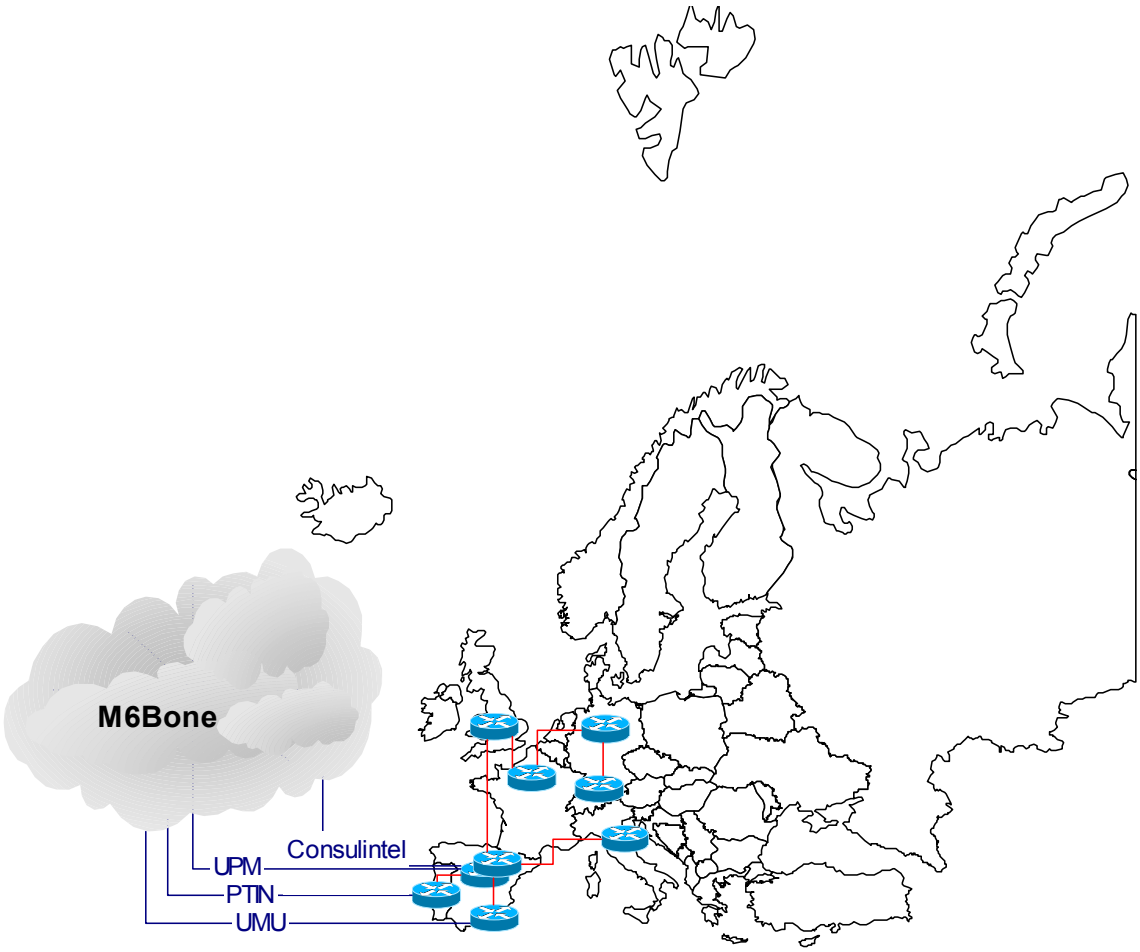

Figure 2-47: Transition Scenario for Euro6IX Connection to M6Bone Network

Second scenario

A RP is introduced inside one of the partners' IXs. In this case, Euro6IX network would have its own RP independent on that one managed by Renater. This scenario is more complex to manage for the presence of two RPs (one in the M6Bone and another one in Euro6IX) but it would permit to analyze some interesting research issues related to the multi-domain multicast routing. During second half part of Y2 the aim is to install at least one RP in one 6IX.

In parallel of this work, the Multicast Interconnection between Euro6IX and 6NET is also considered within this sub-activity, because of it is highly desirable to establish IPv6 Multicast connectivity between both IPv6 networks. UoS is jointly a participant in Euro6IX and 6NET, and it is one of the M6Bone participant sites. UoS proposed to continue to operate its own IPv6 multicast infrastructure along with to:

- Participate actively in the M6Bone.
- Join the Euro6IX and 6NET native multicast infrastructures as they become available during the lifetime of both projects.
- Specific new technology will also be tested as it becomes available, e.g. PIM-SSM and MLDv2 implementations.

Regarding Multicast applications, the A4.1-5 sub activity addresses this mainly working on Digital TV servers. The basis is a web interface where the user will be able to select a TV channel and automatically join a multicast group and watch the TV. The final service should include some kind of authentication.

Along this, also is addressed the Euro6IX participation on demonstration for internal and public trials.

Relevance

Due to the fact that Multicast IPv6 is becoming a demanded and very used service, it is important to allow Euro6IX partners and clients to have a good multicast connectivity. We can state that the relevance of Multicast Evaluations sub-activity is high because:

- Multicast is a key issue for IPv6.
- The use of Multicast enabled applications can improve the use of networks and the development of new services.

Results

By the end of Y2, the work done within Multicast sub-activity includes:

- Gathering information on deployment and tests for Multicast infrastructure and applications.
- A proposal led by UMU and TILAB, where have been identified two possible scenarios for Euro6IX gains M6Bone connectivity.
- A proposal led by UoS to interconnect the Euro6IX and 6NET native multicast infrastructures as they become available during the lifetime of both projects.
- Toward the installation of at least one RP in one 6IX, TID was connected to the M6Bone using an IPv6 multicast tunnel over IPv6 unicast. The most interesting point was that MBGP was used instead RIPng, to route between Renater and TID (http://www.m6bone.net/sites-map.html#fm6bone). Since this step, in the following weeks, Spanish partners will be connected to M6Bone trough TID/MAD6IX.
- Currently TID is developing a Web-Digital TV server. The basis is a web interface where the user will be able to select a TV channel and automatically join a multicast group and watch the TV.
- Participation in 1st Euro6IX public trial during Madrid 2003 Global IPv6 Summit (May 2003).

Future Work

- Install at least one RP in one 6IX following the proposed scenarios (in progress).
- Connect of Spanish partners to the M6Bone through TID/MAD6IX (in progress).
- Interconnect Euro6IX-6NET networks using Multicast.
- Continue improving the Digital TV server as well the video on demand server (unicast).
- Develop of a VoD service with users and billing.
- Develop of a Digital TV service with users and billing.

2.5.2 Multihoming

Partners

UPM, Consulintel, FTRD, UMU

Objectives

The general objectives of the Multihoming common evaluation sub-activity are:

- Analyze the status of the different multihoming proposals inside IETF and other relevant foras and study which of them can be deployed in IXs.
- Identify the way multihoming issues may be improved or even solved within IPv6 network environment.
- Evaluate IX multihoming scenarios based on IX model C defined in WP2, optionally combined with the use of a Route Server. Evaluate also possible address management systems to support IX multihoming services.

Apart from that, UPM specific objectives inside this sub-activity for Y2 are:

- Study the multihoming solution based on RFC 3484 (Default Address Selection for IPv6), and propose an extension that includes the participation of routers or external servers.
- Testing the USAGI Linux partial implementation of RFC 3484, verifying its correct behavior with a varied set of source and destination IPv6 addresses.
- Develop a user space utility (missing in the USAGI implementation), used for changing the kernel label table configuration described in RFC 3484, and providing the related support in the kernel.

Description

Multihoming, which is defined as the connection of a network to more than one IP Service Provider (ISP), is at present a key component of serviced demanded by organizations connected to the Internet.

In general, this sub-activity will deal with the study of all issues related to multihoming, focusing on the proposals being discussed, its application to IX models, and the identification of the ways they may be improved or even solved within IPv6 network environment.

The work will be based on the study of:

- Current IPv4 multihoming practices.
- IPv6 multihoming requirements according to the ongoing work at IETF.
- Impact on the Euro6IX IPv6 architecture, as well the potential solutions.

The aim with this work is to reach a better understanding of Multihoming main issues regarding to Routing table growth, DFZ convergence time increase and IPv6 aggregation principle break. Thus further work is needed on IPv6 requirements (study of the requirement that IPv6 multihoming has to meet) and impact on IPv6 IX (to analyze the impact and the consequences of multihoming on IPv6 IXs, and will discuss the applicability to Euro6IX test-bed).

Apart for the theoretical study of multihoming solutions, some practical work has already been initiated inside this sub-activity. In particular, UPM has concentrated its effort during the first part of Y2 on one of the host based multihoming solutions: The Default Address Selection mechanism defined in RFC 3484. Basically, RFC 3484 describes two algorithms for source address selection and for destination address selection that must be implemented by every IPv6 node. These mechanisms, conveniently implemented and probably extended can be used as the basis of a host based multihoming solution.

IPv6 capable USAGI kernel implements RFC 3484, in particular, what is related to "source address selection by host" solution. This implementation has been tested, verifying its correct behavior according to RFC 3484, and finding its lack of a user space configuration tool that allows the modification of the policy table. Without this configuration capability, RFC 3484 becomes insufficient to manage a multihoming scenario, which needs some kind of distinction between its destination addresses.

UPM has developed some extensions to the USAGI kernel, together with a user space configuration tool that allows the dynamic modification of the source address selection policy table. Entries to that table, which is static in the original USAGI kernel, can be added, modified or deleted dynamically from the configuration tool by means of some kernel “ioctl” calls. Besides, support in /proc has been added, so there is read access to the Policy table through /proc/net/addrselect_label_table.

Relevance

We can state that the relevance of Multihoming evaluations sub-activity is high. The IPv6 addressing architecture allows multiple unicast addresses to be assigned to interfaces, then Multihoming situations will result in more addresses per node. The end result is that IPv6 implementations will very often be faced with multiple possible source and destination addresses when initiating communication, needing in this way a deeply study of Multihoming solutions.

Results

By the end of Y2, the work done within Multihoming sub-activity mainly includes the work done by UPM.

- Gather information about Multihoming issues.
- Implementation of source address selection mechanism described in RFC 3484 based in USAGI's partial implementation. This tool allows the user setting a table (called policy table) relating destination networks (IPv6 address/Mask) and source address to be used when connecting to these destinations. The tool is available at UPM Euro6IX web site.
- Development of a web demonstration scenario based in UPM Looking Glass (http://www.upm.euro6ix.org/cgi-bin/looking-glass-upm-v0.6/ntools.pl) and VNUML (http://www.upm.euro6ix.org/vnuml/). The demo focuses on showing how the solution implemented can be used to solve simple multihoming scenarios like the one showed in Figure 2-48. The demo shows:
    1. The problem in this scenario: The multihomed host is not able to reach a destination due to wrong address selection.
    2. The solution to the problem: A suitable entry in the policy table.
    3. The possibility to test of the solution: The multihomed host reaches the destination, since the source address selected has been set correctly.

  Remote users, through a simple IPv6 enabled browser, are able not only to follow the demo, but also to play with the kernel label table over a more flexible interface. The results of the operations are also displayed

- The addrlabel tool, developed at UPM, has been integrated in USAGI mainstream since 18th August, 2003 snapshot.

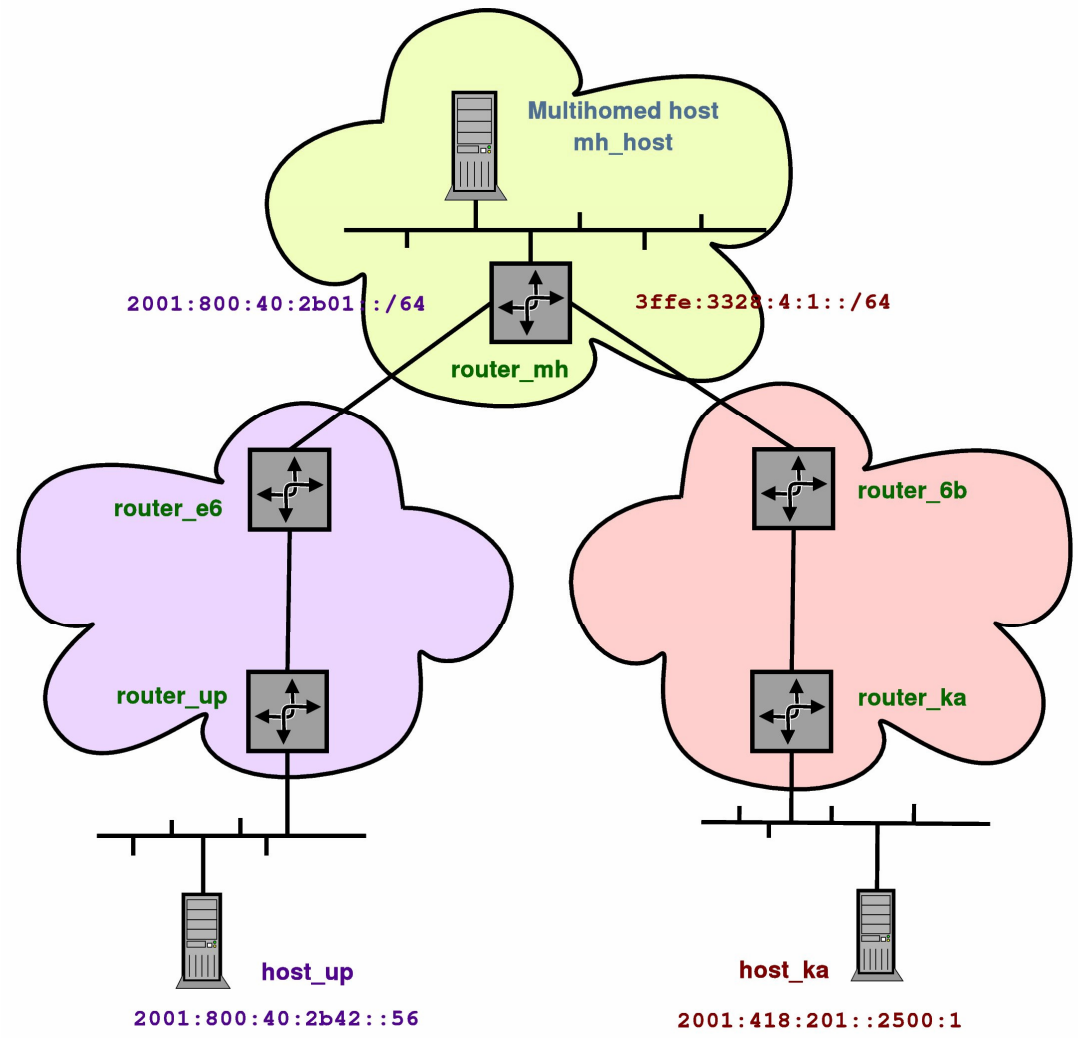

Figure 2-48: UPM's Multihoming Web Demonstration Scenario

Future Work

For the general activities the future work will include:

- Further work related to current IPv4 multihoming common practices, IPv6 multihoming requirements and impact on the Euro6IX architecture.

Related to UPM's activities, it will be concentrate around three main objectives:

- Automate the change in the rules defined in the Policy table, according to the network state or the network administrator discretion.
    1. Using Router Advertisements with an option indicating the source address to be used to connect to destinations. Draft "Default Router Preferences, More-Specific Routes, and Load Sharing" defines an option in Router Advertisements that instruct about the hosts the routers they should use to get certain destinations. A similar option indicating source address to be used with certain destinations could be used.

2. Using a centralized server or router for the Router Advertisements generation. The configuration of the Router Advertisements generation may be static, set by the network administrator, or dynamic, based in the information the routers have about the network state.

- Solve the problem of LANs with more than one router doing advertisements. The solution would be to select the source address to be used, and then use as gateway the router that has announced this prefix.
- Try to solve the scalability problem in the size of the Policy table. One possible solution would be the use of redirects similar to the ones sent by routers to inform hosts about the best gateway to be used for this destination. A similar packet could be defined, informing the hosts about the best source address to use the next time they want to communicate with a given destination.

3. RESULTS ON APPLICATION DEVELOPMENT

This section summarizes the main results obtained in WP4 in the context of activity A4.2, Applications Development, during the second year of Euro6IX project. The subsections below just outline the results of the different sub-activities under A4.2. More detailed information can be found in the Technical Reports associated to each sub-activity (see Section 5).

3.1 Advanced Applications (A4.2-1)

3.1.1 Multimedia Service Provision Platform using SIP

Partners

UPM

Objectives

The objective of this sub-activity is to consolidate the work started in year 1 around SIPv6, focusing on the following goals:

- Study schemes to facilitate **interworking** between ISABELv6 multiconference application and SIPv6 clients.
- Creation of an **ISABEL-SIP Gateway** for interactive participation.
- Use of SIP for distributed **event setup** by means of remote presence or instant messaging applications.

Description

ISABEL is a videoconferencing application that supports different types of services like tele-meeting, tele-class and tele-teaching. As many any other videoconferencing system, ISABEL needs a session protocol to setup media sessions and a conferencing control protocol to manage the conference. Currently, ISABEL has its own protocols to do this task.

The Session Initiation Protocol SIP (RFC 3261) is a client-server protocol used for the initiation, management and termination of media sessions between user agents. Although SIP can be used to create conferences without any extension, that can not be done in an easy way. The IETF (Internet Engineering Task Force) is working on an SIP Conference Framework and it is developing some extensions to use SIP as part of the conferencing control protocol.

Complex multimedia service sessions, such as the ones supported by ISABEL, require advanced features for platform deployment, configuration and operation. The use of SIP for such complex task is not straightforward. SIP was conceived for a much simpler scenario and his application for session negotiation for advanced multimedia service platforms has to be studied in the framework of the Euro6IX network.

Integration of SIP elements with ISABEL components, both with IPv6 support, into the IX structure is a medium term issue. This study will produce results about how complex services provisioning may be supported by SIP and IX elements.

Relevance

The integration of ISABELv6 and SIPv6 is mainly of relevance for the Euro6IX trials, because it allows the access to ISABEL sessions from standard SIP clients, even allowing interactive participation. Besides, it allows also the study of SIP protocol over mixed IPv4-IPv6 scenarios.

Results

During the first part of the Y2, the study of interworking schemes between ISABELv6 multiconference application and SIPv6 has been performed. Fourth areas have been identified to introduce SIP in ISABEL:

- **SIP for conference event management and organization,** that is, to use SIP to create and organize the parallel control infrastructure around an ISABEL event, for example, to detect site presence, to collect session information needed to generate the EDL file, to connect to ISABEL session, to maintain a parallel chat control session, etc.
- **SIP for Conference Media Session**, that is, to use SIP as the media signaling protocol for ISABEL multimedia flows, allowing the negotiation of RTP sessions with external user agents.
- **SIP for Conference Control**, that is, to use SIP as the conference control protocol, using the SIP extensions being proposed at present inside IETF WGs for this purpose.
- **SIP for Interworking**, that is, to use SIP for interworking between ISABEL conferences and standard SIP based videoconference terminals.

Recent work has concentrated on the last point. A detailed study has been work out in order to identify the functionality of an **ISABEL-SIP Gateway**. The identified functionality has been the basis for the implementation of the Gateway. A first version has been developed and demonstrated during Madrid 2003 Global IPv6 Summit public trials (May 2003).

As shown in Figure 3-1, the ISABEL SIP Gateway (ISG) is based on two main components: a modified ISABEL Interactive Terminal and a custom SIP conference server.

The ISABEL Interactive Terminal is able to join a regular ISABEL session and, therefore, it receives and performs the local presentation of all media involved in the session. In addition, this terminal delivers the mixed raw audio stream to the local SIP conference server. It has also been modified to be able to receive an audio and a video flow from the SIP conference server and deliver them to the ISABEL Conference as if they were its own local media streams.

The SIP conference server is a custom built java-based SIP application. It gets the ISABEL session's audio stream from the modified ISABEL Terminal and converts it to RTP format so it can be delivered to standard SIP clients. Unfortunately, this procedure cannot be applied to video, since ISABEL usually presents several video streams at the same time and there is no way to determine which the main or predetermined one is.

Additionally, ISABEL terminals also exchange non-RTP media such as slides, shared applications, etc. so, in many occasions, the speaker's video does not provide enough information to follow the session properly. The provided solution comprises capturing the whole local ISABEL Desktop and converting it to a RTP video flow. The desktop capture process can be either software or hardware based, depending on the available computing resources (software capture is a high CPU demanding task).

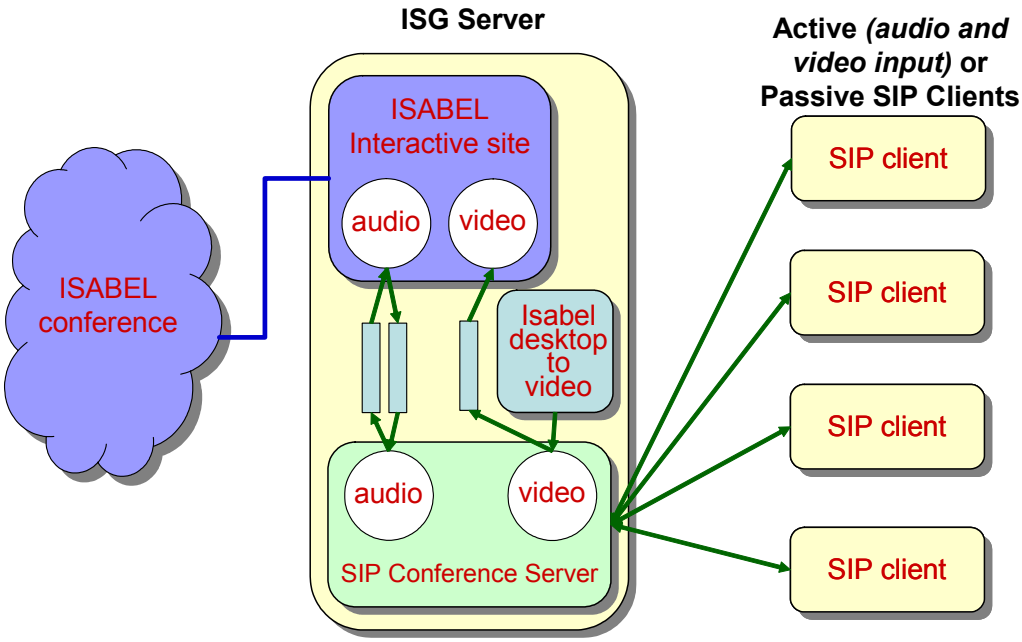

Figure 3-1: ISABeLE-SIP Gateway Architecture

The SIP conference server delivers these RTP flows to SIP clients, and also receives the clients' media streams. Since the gateway is perceived in ISABEL as a regular terminal, just the incoming media from one of the SIP clients can be incorporated to the conference at a time. This selection can be performed dynamically from the user management panel provided by the ISG Conference server. This panel has been designed providing the means for a tighter integration with ISABEL in the future, making incoming SIP-user selection an integrated component in the ISABEL toolbar. After the appropriate format conversion process, the selected user's streams are delivered to the local ISABEL Terminal to be included in the ISABEL session.

The ISABEL SIP gateway is still in an early stage of development, but provides full functionality with a reduced set of codecs. It has been successfully tested with commercial IPv4 SIP clients. Since there were no available SIP IPv6 clients for testing purposes, it was necessary to develop a java-based IPv6 SIP client to test the ISABEL gateway in IPv6 environments. Both the gateway and the IPv6 client were shown in the Madrid 2003 Global IPv6 Summit showroom as part of the Euro6IX demonstrations stand.

During last semester, the java-based IPv6 SIP client has been ported to applet-mode, enabling a simpler and quicker client distribution and allowing clients to follow and/or join an ISABEL Conference from their web browser.

Future Work

Work under this sub-activity will continue during Y3, probably centred around the:

- Enhancement of the ISABEL-SIP gateway, increasing the number of supported media codecs and providing new user management functionalities. More compatibility tests will also be performed as soon as additional IPv6 enabled SIP clients are available.
- Integration between SIP and Euro6IX VPN deployment framework.
- Creation of a SIP framework for ISABEL session setup and management.

3.1.2 IPv6-Capable ISABEL

Partners

UPM, UMU, UoS (tester), TILAB (tester), Vodafone (tester), FTRD (tester)

Objectives

The objective of this sub-activity is to advance the work started in year 1 of fully porting ISABEL to IPv6 and integrating it with the Euro6IX Security Framework. Along Y1 the work focused mainly on completing the code-porting task. Year 2 has been devoted to improve the usability of the ISABEL platform over mixed IPv4/IPv6 scenarios, as well as to integrate new advanced security services on it. This development is relevant for the trials and for gaining experience on porting complex applications to IPv6.

In particular, this sub-activity has addressed the following objectives:

- Enrichment of EDL (sEssion Definition Language) to allow generic platform definition, which allows the integrated definition of ISABEL platform components, IPsec security associations, as well as other advanced services components. The new version of EDL, named XEDL, is aligned with IETF and W3C standards.
- Integration of ISABEL platforms of arbitrary topology with IPsec based VPNs.
- Design of a new management tool that allows participants to subscribe to an ISABEL session and automatically generate and distribute the information needed to manage and start it.

Description

EDL is a key component of ISABEL platform. It is a language used to precisely define all the components participating in a session. However, as it has been experimented in recent trials, the management of ISABEL sessions is significantly more complicated in mixed IPv4-IPv6 scenarios, partially due to limitations of the language.

Besides, the topology used in ISABEL to interconnect the different components has been shown to be difficult to implement in transition scenarios. In addition, the topology complicates the addition of new services like security. Therefore, the first step made has consisted on simplifying it, so that all the network connections in the session follow a tree structure.

After the modifications were made, each station has a well-defined access point which acts as a kind of proxy for all its control and data traffic (Figure 3-2). Each site establishes just two connections (UDP and TCP), addressed to its conference access point. In this way, the connectivity requirements are much simpler, and it is much easier to deploy an advanced network service, such as IPsec or QoS support, inside a session.

A new session description language: XEDL

In order to solve the limitations mentioned, as well as to extend EDL language to introduce new advanced services like IPsec and integrate them inside the ISABEL application, EDL has been redesigned into a new language named XEDL.

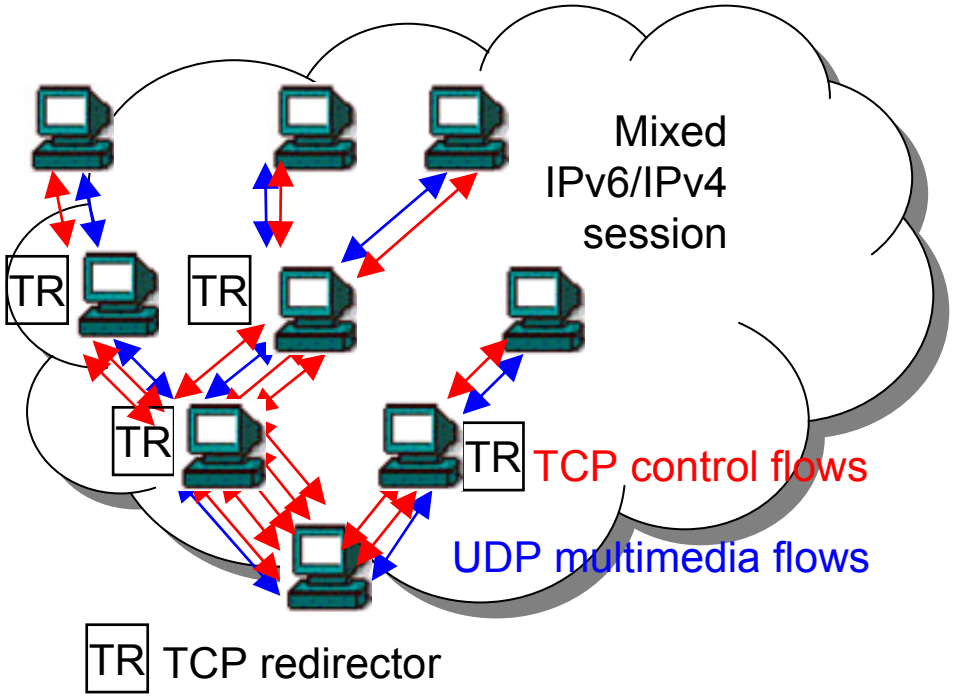

Figure 3-2: Connectivity Requirements after Y2

The main objective driving EDL redesign has been the desire to create a usable and scalable service, where the definition of IPsec security elements and associations, as well as other platform services or parameters of the Euro6IX Integrated Service Framework is possible. Thus, it has been necessary to enrich EDL language. The new EDL is called XEDL, because it is defined in XML in order to align it with other W3C developments.

The current version of XEDL (v1.8) is basically a redefinition of the old EDL language in XML, but enriched with security elements and associations, as well as some changes that make it more flexible. For instance, several addresses are allowed for each "site" (that is necessary to describe mixed v4-v6 sessions in the same file). Also, different network topologies are allowed, not only tree ones (Figure 3-3).

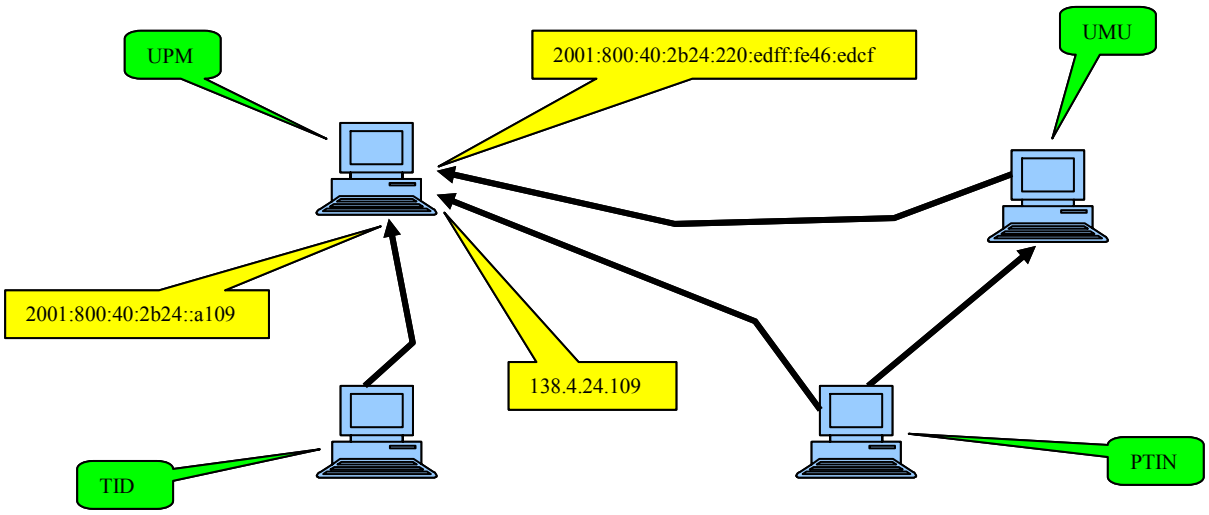

Figure 3-3: Sample Network Topology

Integration of ISABEL an IPsec based VPNs

Experiments integrating ISABEL, IPsec and PKI have been performed in order to assess the maturity of the components to be integrated. A VPN solution using terminal certificates has been deployed to get end-to-end security on each connection, using FreeS/WAN as VPN solution and PKIv6 to provide certificates to terminals. Both IPsec and PKI configuration have been integrated in XEDL as well as in ISABEL application operation. VPN and ISABEL network topologies can be represented using the same XEDL configuration file. This XEDL file can be used to start an ISABEL over VPN session. A VPN management tool developed by UMU is used to start VPNs, and is invoked automatically by ISABEL to deploy them before starting ISABEL. When the ISABEL session has been finished all VPN links are cleared in the same way as they were created.

This scheme protects all data exchanged between the ISABEL components, and so the data confidentiality and the data integrity are guaranteed. The scheme makes use of X.509 certificates to get the authentication of the users. This authentication permits access control and non-repudiation.

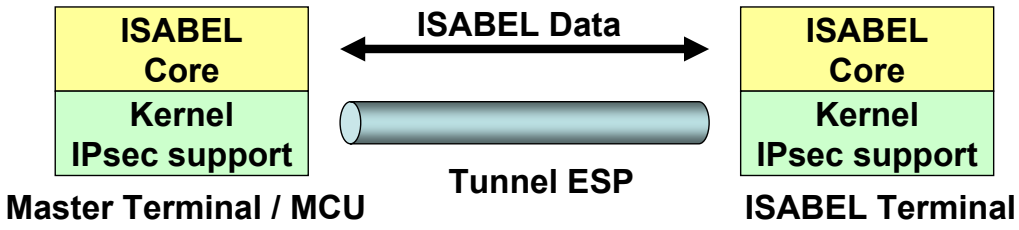

Figure 3-4: ISABEL over IPsec Tunnels

The software used has been release 1.91 of Linux FreeS/WAN with IPv6 support. The implementation of IPsec tunnel mode and the integration in FreeS/WAN was done by IABG as part of the IST projects 6INIT and 6WINIT. ISABEL-4.7 with IPv6 support has been used as well.

A new session management tool

The basic procedure to organize a session consists on generating an XEDL file with all the event information that later will be used by the application to start the session. All sites taking part in the event use external methods -such as email or irc- to send its IPv4/IPv6 address and other network information to the event organizer, who writes manually the XEDL file. This task is especially hard in IPv6 transition scenarios, where IPv6 address are usually autoconfigured, and change frequently.

In order to simplify session management, a new management tool has been designed. The main objective of the new tool is to allow participants to automatically subscribe to the session, centralizing the information about all sites and storing it in the master site. Thus, session XEDL file can be generated automatically and new sites get connected to the session automatically, even without organizer's participation. Apart from that, the tool will also be able to create the web contents needed to announce and coordinate the session.

The remote ISABEL invocation protocol (LIMP) has been redefined in XML RPC to allow a more regular addition of new services. This new protocol has been aligned with existing IETF and W3C standards and should provide a generic invocation service. Interactive sites will subscribe to the session using a URL provided by the organizer, where the site information needed is collected (IPv4/IPv6 addresses, bandwidth, acronym, password, etc).

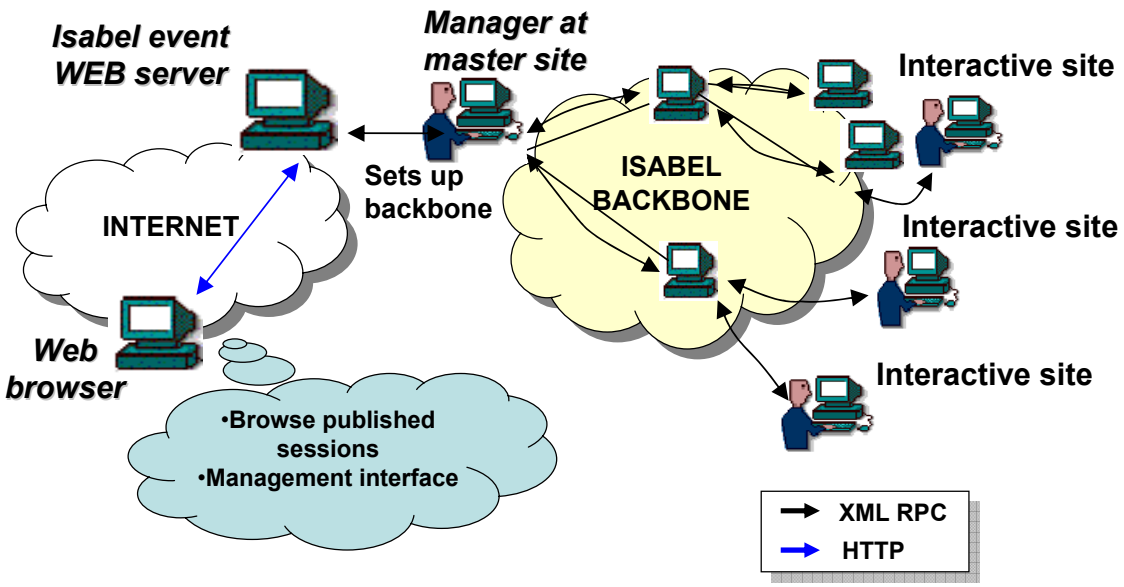

Figure 3-5: Session Management Process

The web site of the event will be updated automatically with the relevant information of each site subscribed to the session, such as administrative information (names, email addresses, organizations), IP addresses, etc. Session information is also stored in the master site in XEDL format, so actual web launch procedure using XEDL is still possible.

Relevance

This sub-activity is based in the integration of an advanced IPv6 capable application, ISABEL, inside the Euro6IX security framework, by aligning the definition of both platforms inside a common standard schema, aligned with IETF standards. This common schema, defined in XML format, will allow not only to define videoconferencing and security platforms, but also to describe other kind of advanced network services such as QOS.

The application is being widely used inside the project, both in trials and meetings, so the improvement of the session management in IPv6 scenarios has a special relevance. It would be very helpful that different partners around the project are able to use the application easily, without suffering from the actual management difficulties caused by the use of the application in a transition scenario.

Results

At the end of Y2, the work done within IPv6 capable ISABEL sub-activity is summarized in the table at Figure 3-6.

The development of XEDL 1.8 has made possible the complete integration of ISABEL with IPsec and PKI. Both platforms are described using the same XEDL file. When ISABEL application is started, UMU's VPN-ISABEL configuration tool is invoked automatically, and VPNs are deployed. Once the VPNs are running, the ISABEL application is started automatically. The deployment of the IPsec platform is transparent for the user, who just has to start the application using a proper XEDL file.

Regarding session management, the first step was to simplify the complex spider's web of interconnection between terminals inside an ISABEL session, in order to have only one access point to the session for each site.

Date	Task	Status
15 June	TCP redirector integrated in ISABEL (connectivity requirements simplified)	Done
31 June	Generation of XEDL 1.7	Done
28 August	Generation of XEDL 1.8. Integration of UMU contributions related to IPsec into the schema.	
2 September	Design of XEDL based management tool	Done
10 September	Remote ISABEL launch protocol migrated to XML rpc (first step towards XEDL based management tool)	
15 September	Integration of XEDL 1.8 with ISABEL (UMU VPNs deployment tool integrated)	Done
30 September	Integration of VPNs deployment at session start up.	Done
16 October	Demonstration of ISABEL + VPN deployment integrated in Euro6IX Y2 review	Done
23 October	Euro6IX 2nd year internal trial: session defined using XEDL 1.8	Done
15 December	First design of ISABEL XEDL based management tool	Done

Figure 3-6: Work Done for IPv6-Capable Sub-activity During Y2

As a second step, the remote ISABEL invocation protocol (LIMP) has been redefined in XML RPC in order to allow a simpler addition of new services. This new protocol has been aligned with existing IETF and W3C standards and should provide a generic invocation service that allows any service component to be launched, such as ISABEL platform components, IPsec components and associations, etc.

Finally, based in the other results of the sub-activity (XEDL 1.8 language, TCP redirector, LIMP protocol redefinition in XML RPC), the design of a new tool has been possible. The objective is to generate automatically XEDL configuration files and web info, so that session management is near automatic.

Future Work

Work inside this sub-activity will continue during Y3, focusing on:

- Implementation of the session management tool designed during Y2. As mentioned, this tool will simplify significantly ISABEL session management. Its implementation is already underway and will be finished during Y3.
- Improvement of XEDL language. During Y3, XEDL 2.0 will be designed with the objectives of aligning it with the main Euro6IX network services framework. The new language will be able to describe and deploy scenarios including different advanced network services like QoS or mobility.

3.1.3 IPv6 Instant and Unified Messaging

Partners

UoS, nGn

Objectives

The general objective of this sub-activity is to implement an IPv6 Unified Instant Messaging system based on jabber protocols, RFC2778 and RFC2779 from the IMPP and IETF working group, and to provide advanced messaging services by making use of the advanced IPv6 services.

For UoS, the Y2 objectives of the IPv6 Unified Instant Messaging are:

- Implement the Jabber based Instant Messaging prototype system over IPv6.
- Expand the functionality and improve the overall performance of the Unified Instant Messaging system by making the best advantages of IPv6 services and Peer-to-Peer architecture.

For nGn, for the first year the objective was to build a simple single-session messaging tool, with IP number or machine's name identification, testing connectivity and transport over IPv6 networks. The objective for this Y2 and Y3 is to get Jabber systems compatibility in order to be integrated in 6UMS (Unified Messaging System over IPv6).

- Enhancement of GUI.
- Development of a set of services and functionalities.
- Jabber systems compatibility.

Description

UoS:

In Y2, UoS has worked in:

- Based on what have been done in Y1, deploy, develop and debug a Jabber based Instant Messaging Server over IPv6 with core functionality provision: messaging, presence management, registration and authentication, roster management, group chat, etc.
- Develop GUI based Instant Messaging Client with friendly HCI through which the IM services can be accessed.
- Debug the interaction between Instant Messaging client and Instant Messaging server.
- Start to use IPv6 advanced services available to provide advanced messaging services, such as mobile messaging, etc.
- Start to extend the client/server model based IM system to hybrid distributed IPv6 peer-to-peer messaging system, as shown in Figure 3-7.

nGn:

During Y1 was developed a simple p2p prototype using IPv6 transport. A very light protocol was designed to implement a p2p message exchange service using XML stream.

During this Y2 application has move towards hybrid architecture in order to include jabber protocols management. Jabber systems are based on client/server model but we will include p2p services when possible to enhance IPv6 features as far as possible.

This development is not a code porting but an adaptation to keep compatibility with proposed xmpp systems and impp theoretical schemes, modifying Jabber schemes to include p2p features, which allow to operate the advantages that the IPv6 protocols contribute to the Internet

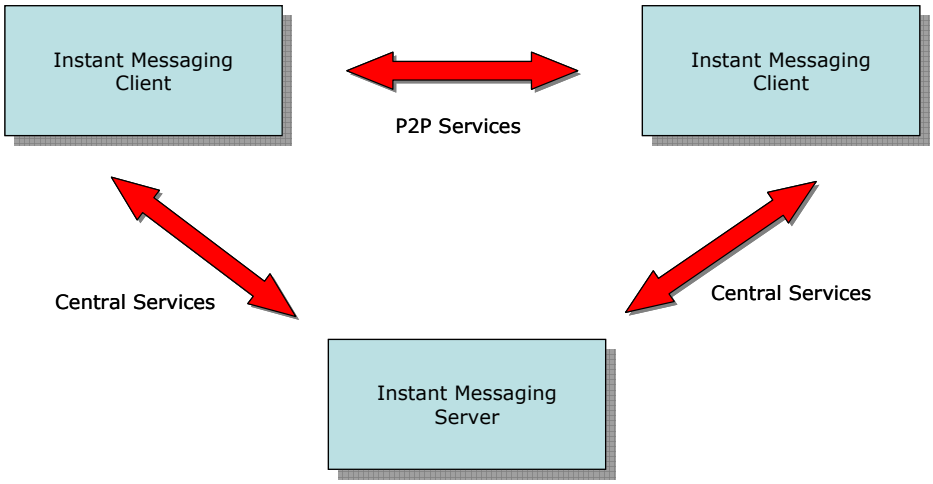

Figure 3-7: Hybrid Model

Protocols used will be like jabber to ensure compatibility with jabber systems and get 6UMS integration. Until now a set of centralized services has been developed using jabber-oriented proprietary protocols (showed later) and complete adaptation and new features will come during Y3.

Relevance

For UoS Unified Instant Messaging System, it is quite relevant to IPv6 networks. Firstly, IPv6 Address Space enables more and more end users and devices to use IM services to exchange messages with their peers, such as the pervasive devices with embedded CPUs and the 3G mobile handsets, etc; secondly, the IPv6 advanced services can be taken advantages of in order to provide advanced IM services and improve the performance of messaging system, such as Mobile (Wireless) Messaging, Secure Messaging, Multicast Based Group Messaging, and Ad Hoc Messaging, etc; and thirdly, the architectures of IPv6 based IXs will ease the deployment of the IM systems, for example, the IM servers (where required) can be connected directly to the Layer 3 IX so that they are independent of the regional ISPs.

For nGn Unified Messaging System, the main objective is to build an attractive application for users to generate an important rate of traffic over IPv6 network. P2P technologies are favoured by IPv6: number of IPs (number of different devices), public IPs (no NAT necessity), end-to-end security, mobility; and Jabber systems are client/server oriented, so an adaptation into hybrid systems should be interesting to IPv6. Instant Messaging is an integrated application into Euro6IX Services Framework: Compatibility with UoS system (Unified Messaging System over IPv6) and end-to-end security service.

Results

So far, UoS has:

- Deployed and debugged the Jabber Instant Messaging server.
- Developed the GUI based Instant Messaging client.

- Debugged the interaction between IM client and IM server over IPv6 so that the following IM services can be provided:
    - User management (register and authentication).
    - Roster management (add/delete friends).
    - Presence management.
    - Messaging (Instant Messaging/Queued Messaging).
    - Out of Band Messaging (Peer to Peer Files Sharing via HTTP URIs).
    - Group Management (Create/Join/Leave Groups).
    - Group Chatting (Optional User Relayed Group Messaging Delivery).
    - Peer to Peer Voice Chatting (Signalling using Instant Messaging).
    - Wireless Instant Messaging test.

So far nGn has implemented:

- Registry service and authentication service. Using a central database, a client/server model. User is identified in the system with a single nick, without IP or machine's name reference.
- Roster management service, to manage user's contacts list.
- Presence service. A hybrid service is used, with p2p notifications every time a change in the status occurs, and with a central subscription service. A peer is automatically subscribed to presence service for every peer found in its contacts list.
- Messaging service. A p2p service to exchange short text messages.
- User's Search Service, to find possible contacts and select them only by their nicknames.
- Improved GUI.

Future Work

In the future, UoS plans to:

- Implement the wireless (mobile) Instant Messaging system by making use of mobility services provided by IPv6.
- Implement the end-to-end secure messaging.
- Improve the performance of group messaging by deploying and using of IPv6 multicast services.
- Expand the existing client/Server based IM system to a hybrid distributed peer to peer messaging system (agent based).
- Develop other advanced messaging services, such as ad-hoc messaging, etc.
- Integration with nGn Unified Messaging system, as shown in Figure 3-8: IPv6 Instant and Unified Messaging Integration
- .

The future work by nGn will focus on:

- Jabber identifiers management.
- Different jabber resources management.
- Jabber Compatibility for clients.
- Dial back protocol between servers.
- P2P File sharing service.
- Security.

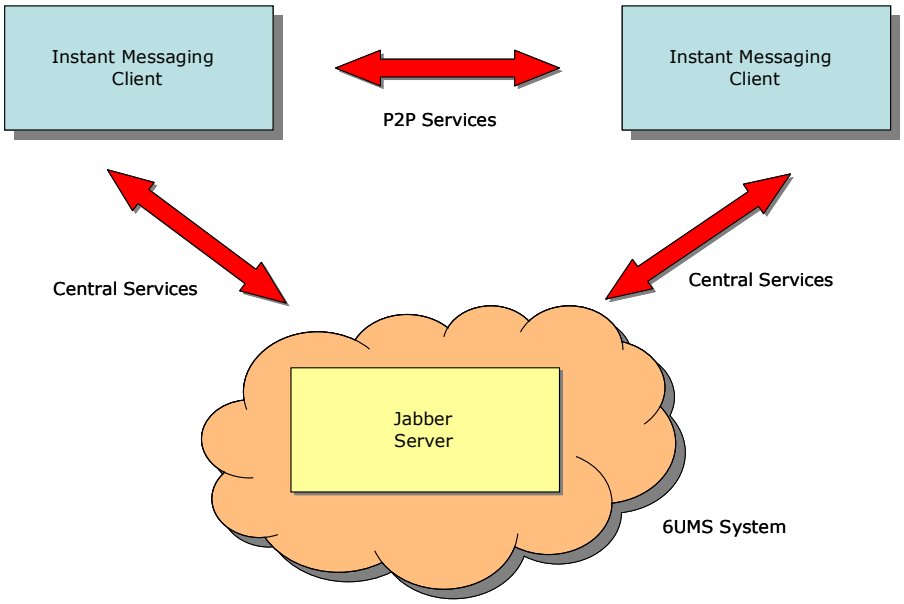

Figure 3-8: IPv6 Instant and Unified Messaging Integration

3.1.4 Groupware Applications

Partners

UPM, UMU

Objectives

The objective of this sub-activity is to advance the work started in year 1 of porting and consolidating in IPv6 several groupware tools integrated into the Euro6IX security framework, which are relevant for the trials and for gaining experience on porting to IPv6. This sub-activity addresses in particular the following developments:

- Integration of the AGWS set of Web based groupware tools with the Euro6IX Security Framework and completion of porting to IPv6.
- Usage of the VNC tool for creation of an IPv6 Web Antenna, which allows high quality streaming of ISABEL sessions over standard Web browsers.

Description

The integration of AGWS with the Euro6IX security framework started in 2Y1, where authentication of users based on certificates and the PKIv6 service was integrated, avoiding the need of user identifiers and passwords. During Y2 a second step towards the integration of AGWS, ISABELv6 and the Euro6IX security framework has been performed, where configurations of ISABEL including security (based on EDL and policy definitions) have been included in AGWS in order to allow new ISABEL end-to-end security requirements to be launched from AGWS.

This development has been validated with a first trial set up where two secure connections for an ISABEL terminal have been started by an AGWS user. The first one is between the terminal and the session master, and the second one between the terminal and the ISABEL MCU (FlowServer). Both connections are insecure. The section "3.1.2 IPv6 capable ISABEL" shows the solution to get end-to-end security in these connections using IPsec VPNs and certificates.

In addition, the PostgreSQL database has been updated in AGWS to version 7.3.3 to include IPv6 support in the communication with the data base. The previous porting did only include support to IPv6 in the front-end for Web access.

The ISABEL Web antenna is a high quality VNC-based streaming application that allows users to follow an ongoing ISABEL Session over a standard Web browser. The application makes use of two Java applets. The first one displays the ISABEL desktop on the browser and the second one reproduces the audio of the session in the receiving terminal.

The application architecture of the ISABEL Web Antenna is depicted Figure 3-9. The ISABEL Antenna Server is a gateway between the web clients and the ISABEL Application. It receives the ISABEL Session's audio and video streams, performs flow mixing and format conversion and delivers them to the passive clients in an appropriate format. Its main components are:

- A modified **ISABEL Interactive Site**. This site receives the session's media streams and performs their local presentation. It has been modified to deliver the mixed raw audio stream (PCM) to the local Web Antenna Audio Server.

- The ISABEL Web Antenna **Audio Server**. This server receives the local ISABEL mixed audio stream, performs the appropriate format conversions (from PCM to GSM) to adequate it to Internet Transmission and delivers it to the applet-based ISABEL Web Antenna Audio Clients.
- A **VNC Server**. ISABEL presents simultaneously several visual components (videos, slides, shared applications, etc.) and, since their nature, number and size varies dynamically during the conference, it is not possible to arrange a pre-defined selection. Therefore, the whole ISABEL desktop needs to be delivered to clients, with the lowest possible bandwidth cost but with quality enough to allow users to follow video and slide presentations. This makes a Desktop transmission system such as VNC particularly suitable for this solution.
- A **Web Server**. The application's web pages and applets need to be served. Since full IPv6 support is required, the Apache2 web server was selected.

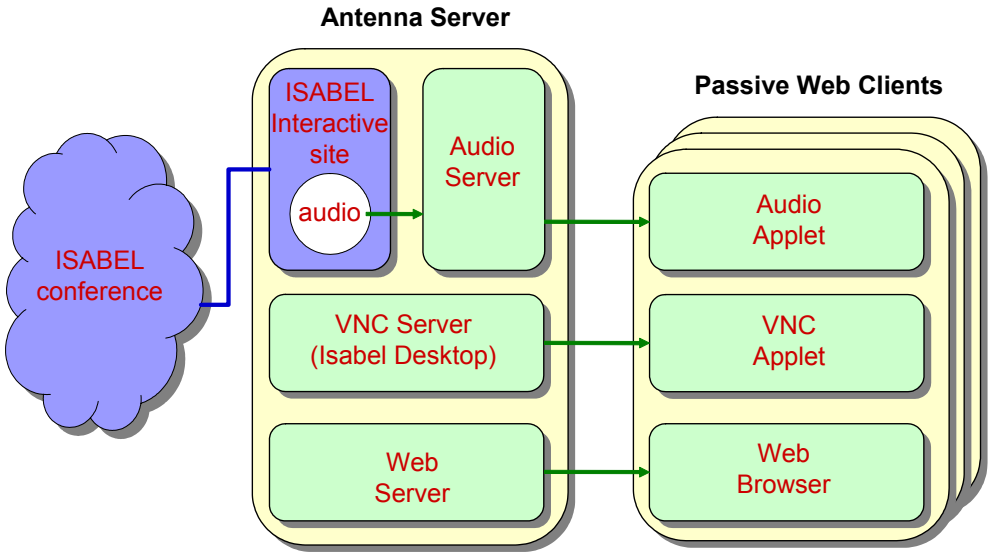

Figure 3-9: ISABEL Web Antenna Architecture

The Antenna Web client retrieves the media from the server and presents them to users in such a way that the ISABEL Session can be followed. It's an applet-based Java application that can be accessed using any java-enabled web browser on any platform. Its main components are:

- ISABEL Web Antenna Audio Applet: Receives the audio stream provided by the Audio server and reproduces it locally. Since this service must be platform independent, audio reproduction is accomplished through the standard javax.sound package included in Sun's JRE. All the necessary codecs to perform format conversion are provided in a jar bundle that is downloaded along with the applet.
- VNC Applet: This standard VNC applet presents the system desktop shared by the VNC server.

Relevance

This sub-activity will provide valuable experience on porting to IPv6 complex applications made of multiple heterogeneous components. Moreover, ISABEL Web Antenna is a key application being used to support the trials, allowing a high quality streaming over IPv6 of ISABELv6 sessions to standard java-enabled web browsers. Finally, the integration of the groupware applications migrated in this sub-activity into the Euro6IX security framework will contribute to test and mature it.

Results

Integration of AGWS, ISABEL and FreeS/WAN to get end-to-end security in all user connections. The end-to-end security is based on the use of PKIv6. AGWS users can define an ISABEL session in an EDL file (with ISA extension) describing the session components (master, flowserver and users). This extension links the user browser to a script starting IPsec and ISABEL. When ISABEL downloads an EDL file, ISABEL and IPsec will start automatically using the certificate user.

The ISABEL Web antenna has been successfully tested in several internal and external events, some of them with a vast audience such as, for example, the Madrid 2003 Global IPv6 Summit.

The main results achieved may be summarized as:

- Development of a complete java-based audio subsystem that allows the broadcast of ISABEL's audio in an efficient way with low bandwidth consumption.
- IPv6 porting of TightVNC Server version 1.2.8. This particular version was selected because it includes view-only password support, which enhances server security.
- IPv6 experiences and code development in Java.

Future Work

The complete integration of AGWS, ISABEL and the security framework needs the new version of EDL, which will allow the storage of platforms definitions integrating ISABEL components and security associations. Presently, the user certificates must be located in two physical sites, the certificate store of the browser and the IPsec directory, and the user has to manage both certificates. In the future, the certificate management must be made transparent for the user.

The porting of AGWS virtual rooms to IPv6 is also still pending. By now a first version of the ISABEL Web Antenna has been produced, but there are still the following developments under consideration:

- IPv6 Support in Windows: Since there is no IPv6 support for Java in Windows systems yet, the ISABEL Web Antenna can only be accessed in Windows from IPv4. Support to IPv6 in Windows will exist in the next JDK release, so upgrade will be necessary.
- User interactivity: Currently, web clients can only follow the ISABEL Session passively, that is, they can receive media from the conference but they can not send media to it. It should be possible to include mechanisms that enable web clients to send media to the ISABEL session, although this kind of solution might reduce platform independence or require the installation of additional Java components, such as the Java Media Framework (JMF).
- Independent Component Video transmission: The whole ISABEL Desktop is distributed to clients using VNC. In order to reduce the bandwidth consumption, the most significant visual components can be transmitted instead of the desktop. Further study is required to find appropriate mechanisms to deal with ISABEL's interaction mode switching.

3.1.5 IPv6 Sharing Information Tool

Partners

TID

Objectives

The objective for Y2 is to finish a complete, functional version of the P2P system, with all the basic features of any P2P software. The server will be fully functional and it will require little modifications, if any, in future versions.

The client is divided in two parts: a daemon running in Linux that performs all the search and transfers functions, and a Graphical User Interface running in Java that connects to the daemon when needed to show the status of the client and perform new queries. Both parts will be fully functional, with basic features.

More in detail, the functionalities that each part will include are:

- Server:
  - A complete server with searches using substrings and hashes.
  - Capable to attend the queries in TCP, UDP Unicast or UDP Multicast.
- Client daemon:
  - Client capable to connect and interact with the GUI.
  - Capable of sharing files automatically.
  - Performs searches and looking for other peers to download the selected files.
  - Capable to auto-resume the files if the connection with a peer is lost.
- GUI:
  - The basic operations supported by the GUI are provided by the base client, like download, upload, search, …
  - The graphical client sends the operation type and the base client executes it with the appropriate arguments. For the communication parameters like IP address of base client, port to connect, password and the delay value for the traffic control refresh can be specified.
  - GUI includes a experimental text-chat between peers in the contact list.

Description

The main change in the description of this sub-activity is the dual client technology, using a daemon and a GUI.

The core of the client, that makes all the search/transfer work, is a Linux application. It runs as a daemon (no console input) in a Linux box. It can work independent from the GUI.

The GUI is a Java application that runs in most of Java Virtual Machine 1.4, including Linux and Windows. The GUI connects to the client daemon to request searches, select files and checking the progress of the transfers. The communication between client daemon and GUI could be over IPv6 or IPv4, depending if you are running it over Linux and Solaris, or over Windows (not IPv6 capable in Java). Figure 3-10 and Figure 3-11 show the GUI appearance.

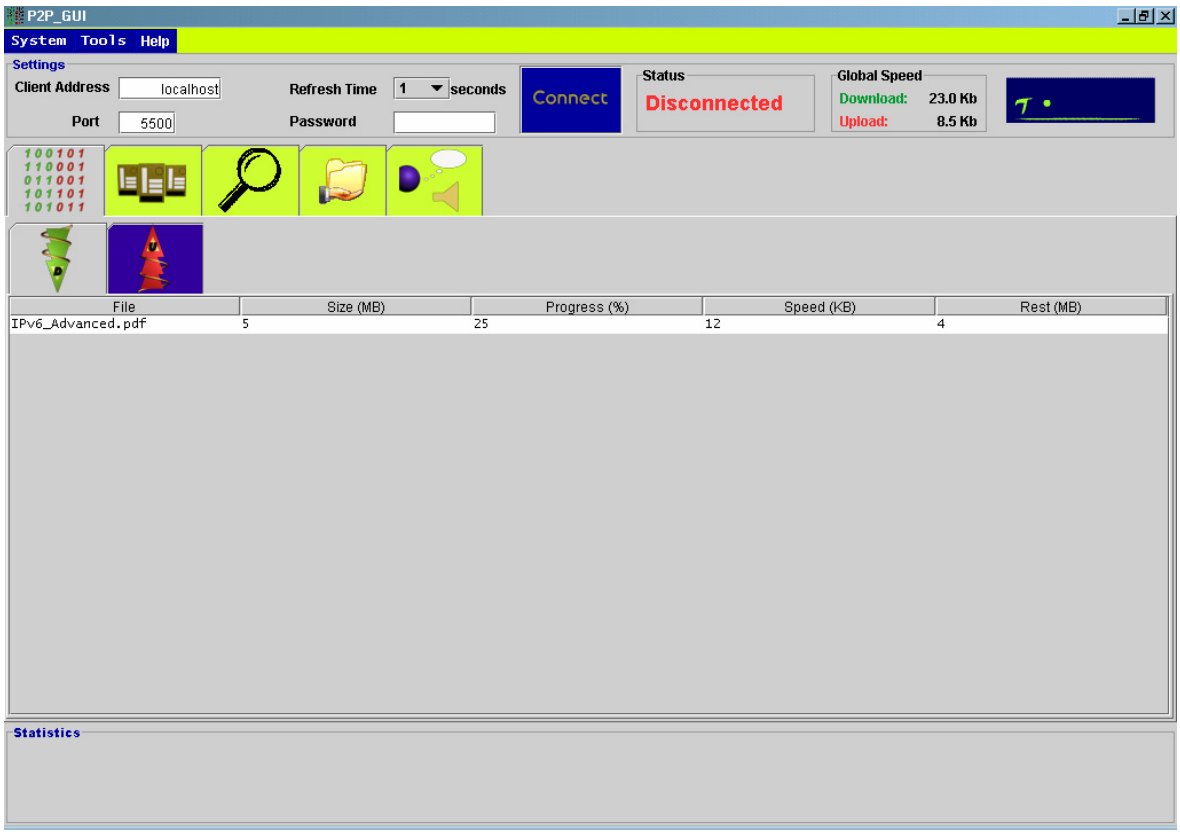

Figure 3-10: General Image of the GUI

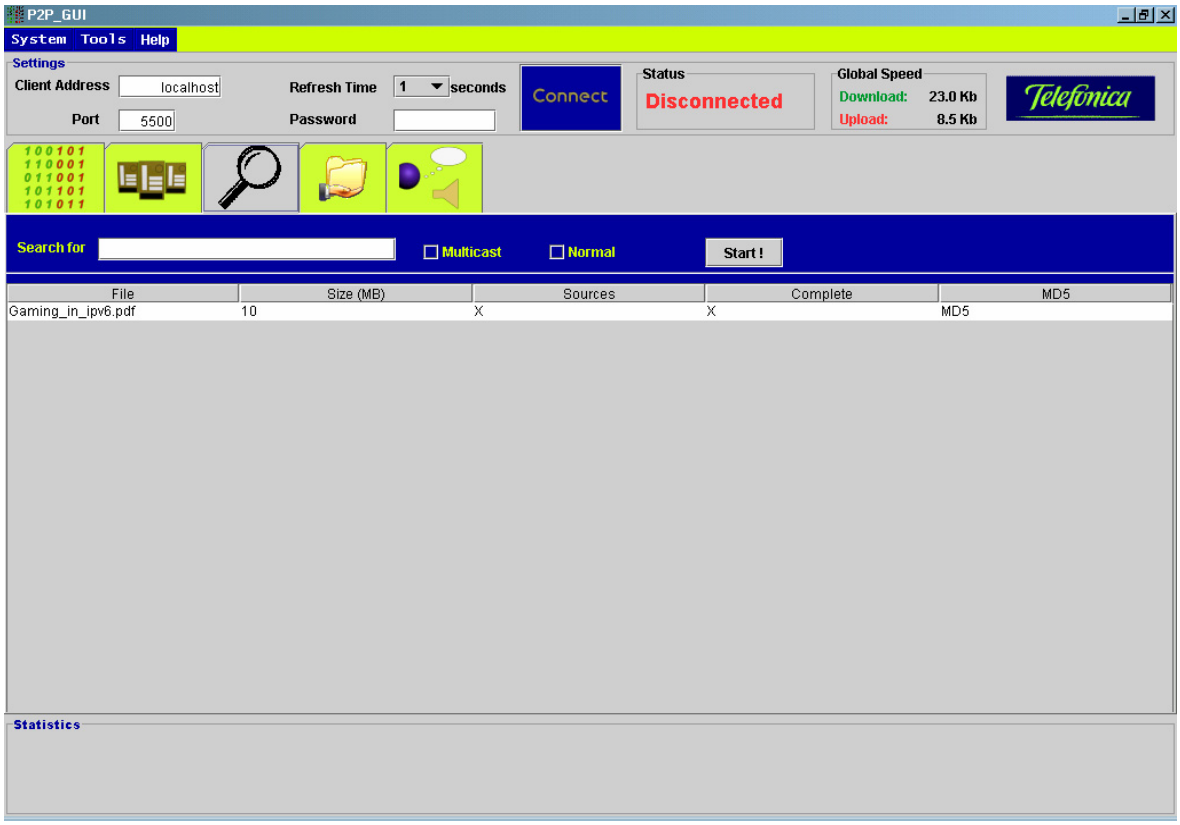

Figure 3-11: General Image with Search Option Selected

This architecture allows the GUI to connect from any place. It is possible to install a client daemon in the IX, where a lot of bandwidth is available, and control it from any computer using the GUI. Other possibility is to connect from a mobile device, like a laptop, maybe from a WLAN access.

The transferred files are kept in the daemon computer. Of course, the daemon host and the GUI host may be the same, acting as traditional one-host software. Figure 3-12 shows the application architecture.

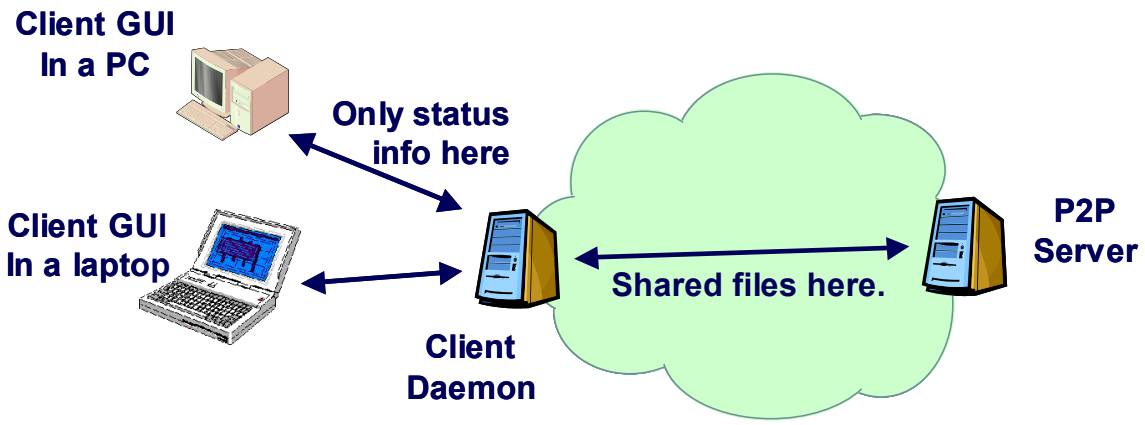

Figure 3-12: Connections between Client Daemon and GUI

The protocol used in all the communications is an adaptation of the Napster protocol, and it is explained in detail in the associated Technical Report.

Relevance

Even though there is a lot of P2P software in IPv4, there is no software of this kind available in IPv6. P2P software has been a great success in the Internet and it would be very interesting to have it with IPv6 as well, as a final costumer service.

The inclusion of Multicast searches is an important new feature, unavailable in similar IPv4 applications. From the IX point of view, the server should be placed near an IX as a high-bandwidth point.

Also, it would very interesting to share public documents or big, public files in a P2P basis, and the IX would be the perfect point of the network. Due to the remote GUI capabilities, the server can be a computer without physical access.

Another advantage of IPv6 is an easy way to establish priorities in the transfers. This will be discussed further in "Future Work" section.

Results

Until this moment, the server has been almost completely developed. The daemon and the GUI are being developed in parallel, and both are in an advanced state.

The partial result reached until October was presented in the project review demo room. The GUI with a simulation was shown and a simple command-line search engine was developed in the client to perform queries in text-mode.

By the end of the year, a complete version of the client is available, capable to transfer files between the clients and with full interaction between GUI and daemon.

Future Work

Once the basic features are implemented, there are lots of new features that can be implemented.

One very interesting new feature to implement is a priority transfers. The idea is to give more priority to those peers nearest to us in terms of network topology. Thanks to the hierarchical structure of IPv6 networks, the peers with more matching network bits will be nearer. This is not possible to do in IPv4, where assignments are random-like.

The client will be able to choose the nearest peer, or the n-nearest peers, to transfer from, freeing as much external traffic as possible. This priority system will be enhanced to allow more priority parameters, giving flexibility to the choosing algorithm.

3.2 Network Management and Operation Tools (A4.2-2)

3.2.1 Magalia

Partners

TID

Objectives

The general objective of this sub-activity for Y2 consists in the addition of two major improvements to Magalia:

- Add graphical edition capabilities to the graphical interface (Xges).
- Creation of a mechanism to share information between IX.

Description

During this year the work in Magalia has been focused in the addition of new features, the graphic interface edition support and the shared management support.

Magalia is a management environment developed in the context of Euro6IX project. The first version of Magalia was released without map edition support and without any mechanisms for sharing information between IXs.

Graphic edition support

Because of the lack of graphical edition capabilities, the map files were based on tkined[1] format. The edition of network maps was done using this tool as a provisional solution.

This year TID has developed a graphical edition support for Xges, so that now tkined package is not needed any more.

The edit support for Xges has the following features (Figure 3-13):

- Edition permissions based on AAA.
- Popup menu based operations.
- Menu of disposable icons.
- Uploading/Downloading network maps from the server.

Shared management support

Magalia has been designed to work in cooperation between different organizations. This is a very sensitive tip because organizations are very reluctant to grant access queries to their equipments.

The Magalia shared management support is a passive mechanism to share information of the whole Euro6IX network.

[1] *tkined* is part of the well know Scotty management environment

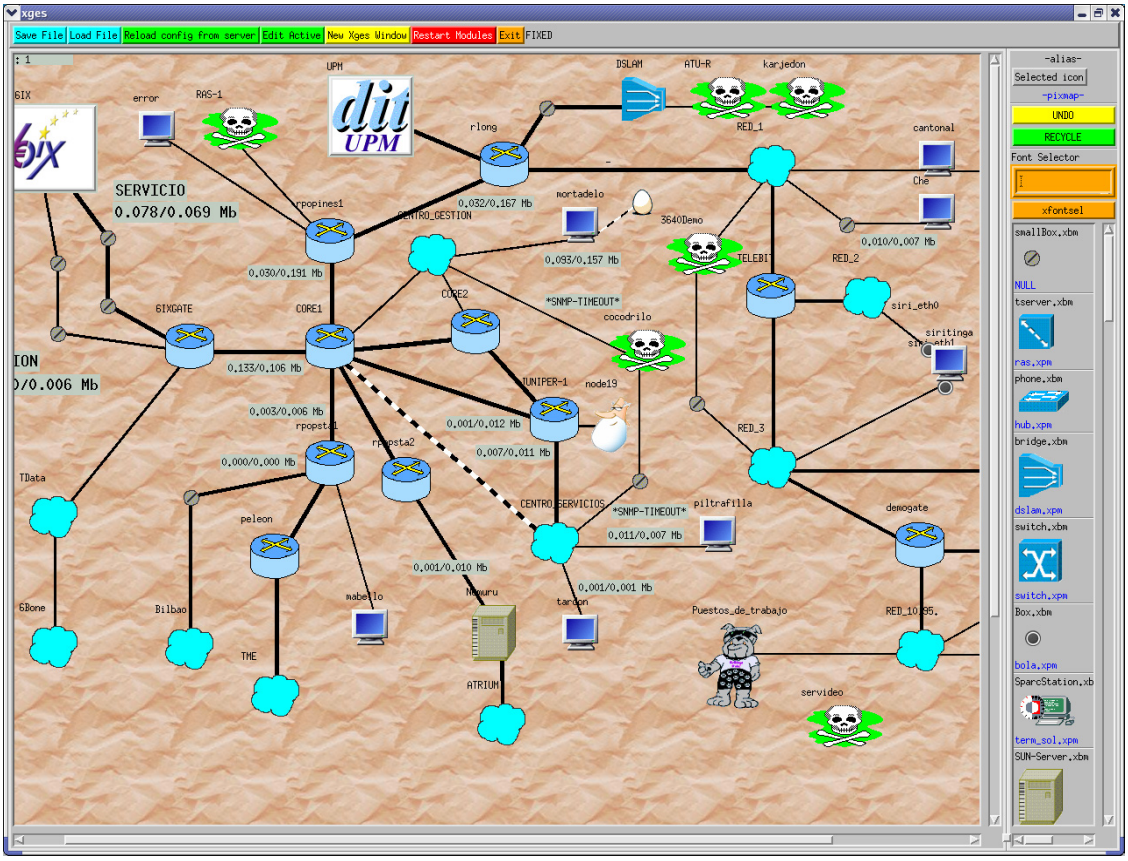

Figure 3-13: Xges with Edit Support

The main features of this sharing mechanism are:

- Each organization decides what information can be delivered to the network.
- There isn't anything like a root server, the system is fully distributed and the organizations must negotiate the relations with others in order to establish a shared management network.

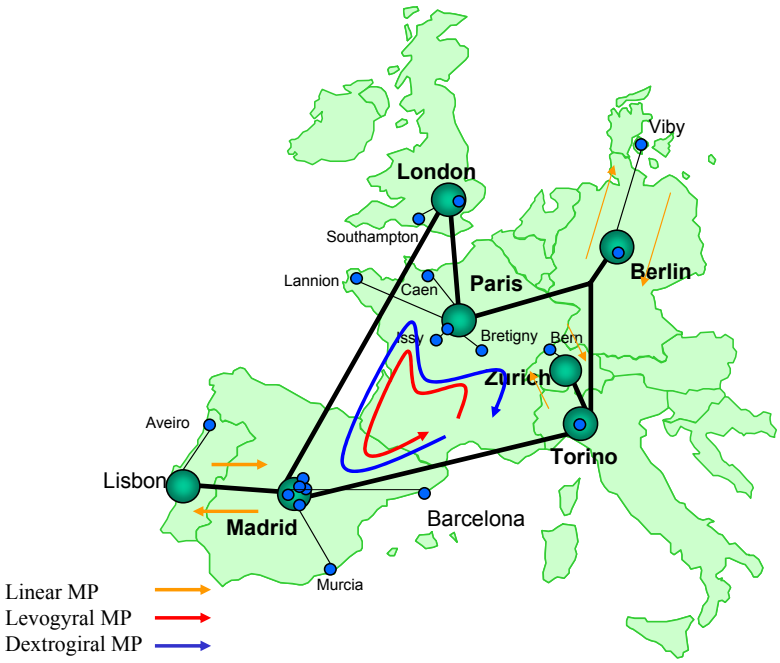

Figure 3-14: Message Paths Propose for Euro6IX

The shared management network is called MSIP, (Magalia Shared Information Protocol), and supports the following features:

- UDP unicast based messages mechanism, extendable to multicast in the future when multicast will be available on the network.
- A specific message format developed for MSIP, (The specification of this format is public and open to Euro6IX members).
- Simply spoofing message protection based on MAC-SHA1 and a shared secret.
- Moderate bandwidth usage in the supervision of big networks, based on message forwarding from one organisation to another.
- Virtual management networks, called MP, (Message paths), with ring, meshed and segmented topologies.

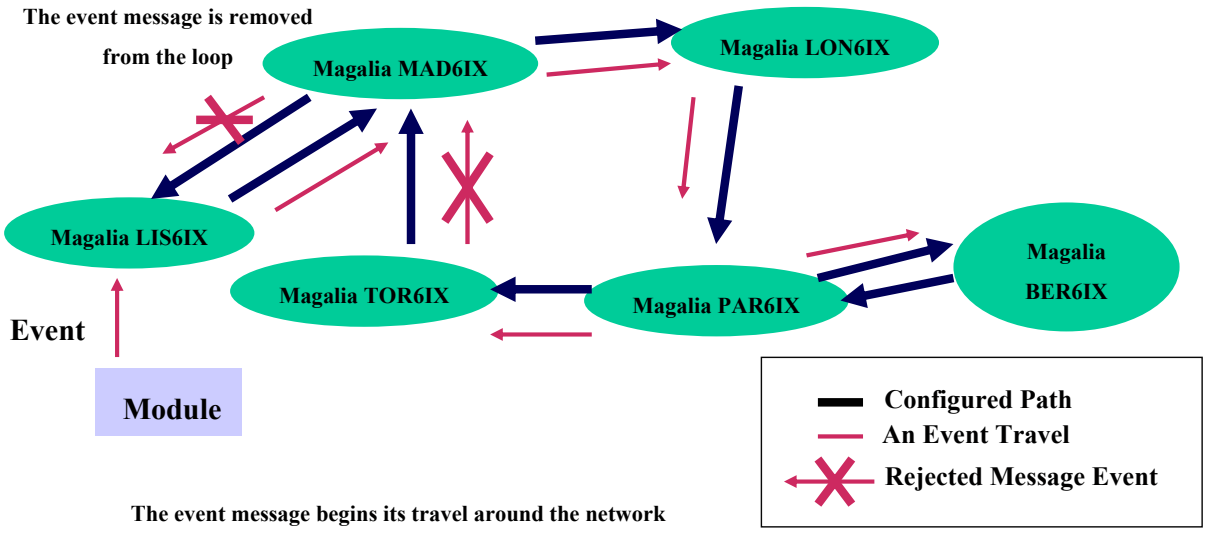

Figure 3-15: Travel of an Event and Forwarding of this Event

In the figure above it can be seen the travel of an event and its forwarding to all the Magalia kernels.

Relevance

Magalia is important for the network, because is the first experience of shared network management between different organizations.

Because the differences between IPv6 and IPv4, a specific management tool is needed, that is the scope of Magalia

Results

All objectives has been reached successfully, Magalia has been installed in UK6IX, BER6IX, MAD6IX, TUR6IX, University of Southampton, UPM and UMU.

All "Magalias" installed in the 6IX nodes are delivering the full state of Euro6IX to the whole network actually.

Anybody with Xges installed can consult the Euro6IX state network in real time logging at TID node:

- **Server:** [mortadelo.tid.euro6ix.org]: 4444

- **User:** Euro6IX_PUBLIC
- **Password:** public

The nodes with Magalia and MSIP installed and configured can consult the same information connecting to its local node instead the TID node, (in this case the user and password can change depending of the local configuration).

Future Work

- Fixing of bugs reported by the members involved in the Magalia trial.
- Enhancements in the security of Magalia and MSIP.
- Implementation of a MIPv6 supervision module.
- Implementation of Alarm and log Modules
- Implementation of a TOPAZ to MAGALIA interface module.
- Study and implementation of network resource requirement modules.
- Enhancements in the Magalia configuration procedures, to obtain an easier configuration.

3.2.2 Topaz

Partners

TID

Objectives

The main objectives for Topaz during Y2 are:

- First stable, tested and debugged version running at TID labs.
- Definition of detection logics using a graphical interface.
- Alerts delivery to Magalia system.

Description

Topaz architecture has been debugged and improved in order to achieve a high efficiency in detecting alerts in Topaz Sensors System and the way these alerts are processed and shown in Topaz Management Console.

A graphical interface has been developed to make easier the detection logics definition. A set of rules and a list of data managers provide this functionality.

In addition, the Alerts Manager module in the console has been modified to forward alerts from sensors system to Magalia system.

Topaz improvements

Regarding to previous work, the Connections Manager module has been deleted since TCP protocol has replaced UDP protocol in the communication between distributed components, i.e., sensors and console.

Modules are fully independent at each distributed component, so the re-starting of one of them doesn't need to synchronize with the rest of them.

The messages interchange between the different modules that compound the application are exclusively between client and server module pairs. As a consequence, the Messages Centre in the console is unnecessary.

The new simplified architecture is shown in Figure 3-16.

Detection logics graphical interface manager

Writing of XML files to define detection logics requires quite time and specific knowledge about XML syntax and DTD (Document Type Definition).

Two windows-based dialogs have been developed to make this task easier, the Rules and List Managers, one for each XML element that is possible to define in detection logics, as shown in Figure 3-17 and Figure 3-18.

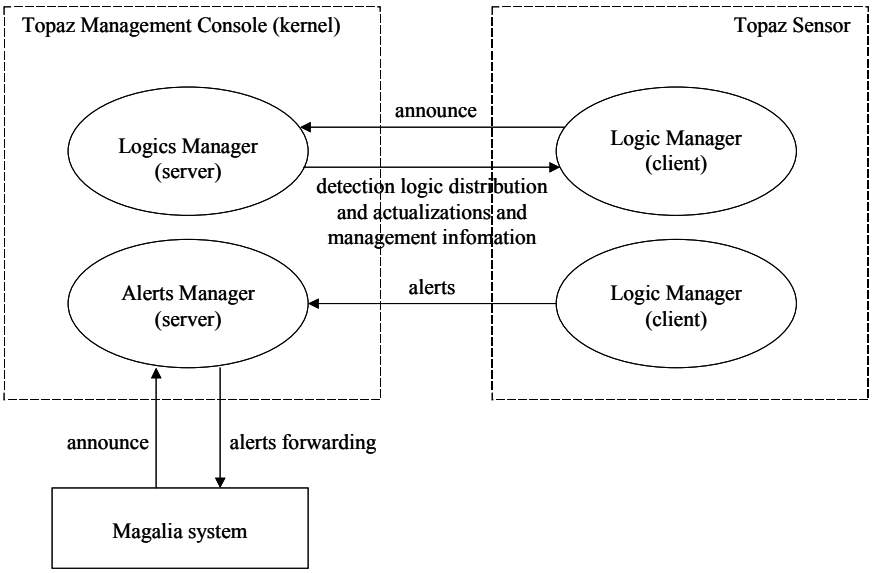

Figure 3-16: New Architecture of Topaz

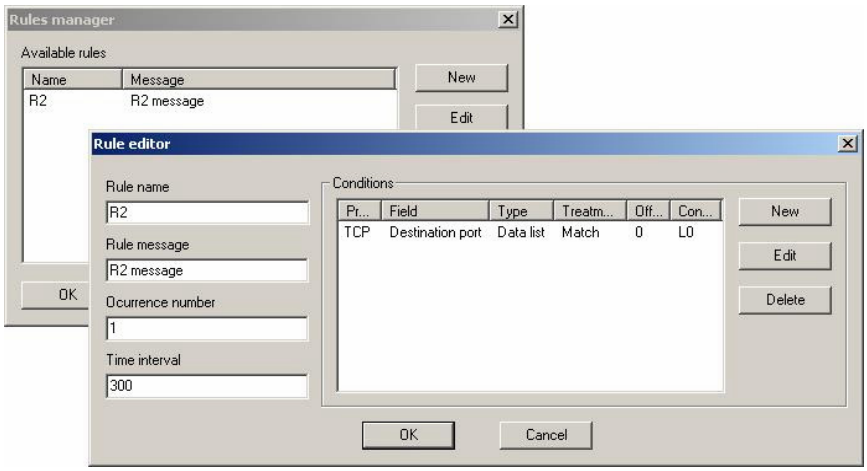

Figure 3-17: Rules Manager

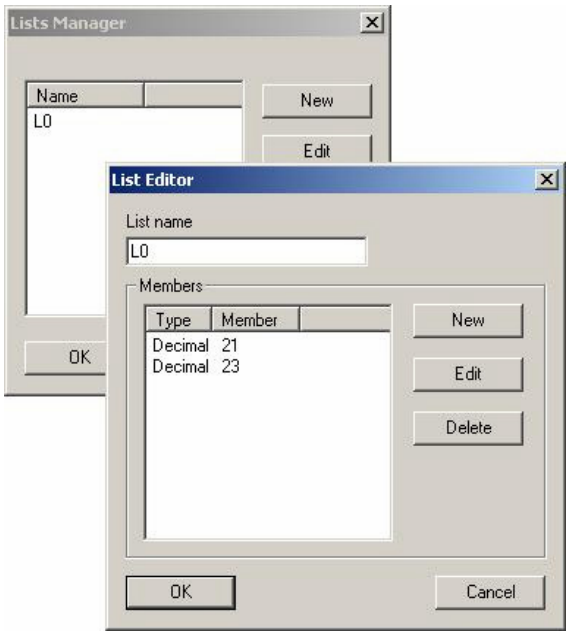

Figure 3-18: List Manager

Magalia alerts delivery

Alerts from sensors system are now forwarded to Magalia system in Alerts Manager module implemented in the console.

The alerts messages format is not modified, so the Magalia module responsible for the reception of alerts delivery must establish a correspondence between Topaz and Magalia messages format.

Communication between Topaz and Magalia system is shown in Figure 3-19.

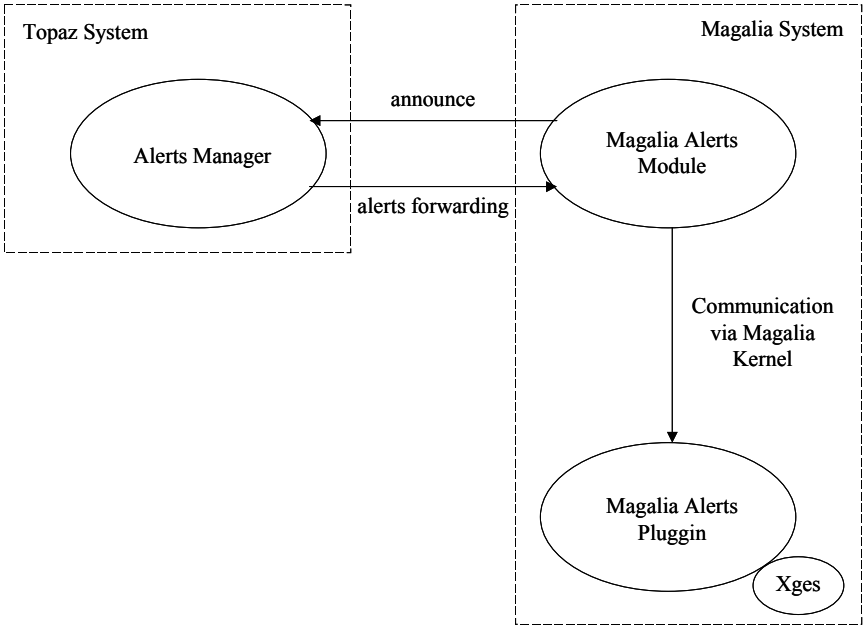

Figure 3-19: Communication between Topaz and Magalia System

Relevance

Topaz is an application to secure Euro6IX networks. Its relevance for IXs architecture and IPv6 is high, as this task implements ways of control and operate Euro6IX network and presents innovate points of view in the way these applications are implemented nowadays for IPv4 networks. Topaz has collaborated during this year with sub-activity A4.1-3 (Security and VPNs in IPv6 networks) to identify the attacks and rules in detection logics.

Results

First stable, tested and debugged version running at TID labs, including fully functionality to detect attacks, define detection logics using a graphical interface and alerts delivery to Magalia system.

Future Work

- Continue collaboration with A4.1-3 to define IPv6 specific vulnerabilities-based attacks.
- Definition of sequence-based attacks.

3.3 IX Support Tools (A4.2-3)

3.3.1 Route Server

Partners

UPM, UMU

Objectives

The objectives for Y2 around Route Server sub-activity are:

- Complete and improve the IX emulation tool developed during Y1.
- Implement IPv6 Route Server functionality in Quagga routing daemon implementation.
- Study and emulate Router Server based IPv6 IX scenarios, as well as the IPv6 IX based address allocation schemes proposed in WP2.

Description

This sub-activity has to do with the Route Server functionality found on IntereXchange points (IX) and its application to new IX models for IPv6, especially in relation with the new IX based address assignment schemes.

Route Server comes up as a solution to IX scalability. It centralizes the BGP peerings made among ISPs connected to an IX, changing the mesh topologies typically found on IXs to a star topology. ISPs only peer with the route server, delegating on it most of the functions (filtering, routing policies, etc) the usually do on their routers.

During Y1, the following results were achieved:

- Collect and study information about Route Server functionality and its advantages and drawbacks.
- An IX emulation tool was developed, in order to emulate complex IX scenarios made of tenths of BGP routers inside one or more Linux system. At the end of Y1, a first alpha version of the tool was available. However it lacked the necessary flexibility and facility of use to allow the emulation of IX scenarios being defined.
- Study and test of IPv6 route server implementations as well as routing policy databases and associated tools. A first survey of implementations was made (mainly based on open source code) and zebra routing daemon was selected to be tested over laboratory test-bed.

All these results were described in detail in the Euro6IX Technical Report TR4.1A.10 named “IX Support Tools”.

Relevance

The relevance of this sub-activity in the context of the project is very high, as it addresses one of the main objectives of the project, which is the study of novel IX architectures and addressing models for IPv6. Besides, the emulation tool developed inside this sub-activity has been found very useful in other sub-activities to facilitate the testing of new services or applications, for example to test IPsec implementations, or to demonstrate other developments like multihoming solutions.

Results

Along Y2 several important results have been achieved:

IX Emulation Tool

The development of the IX emulation tool initiated during Y1 has continued. The tool, named Virtual Network UML (VNUML)[2] is based on User-Mode-Linux (UML) virtualization software and allows the emulation of complex IX scenarios made of tenths of nodes inside one linux machine with possible connection to external equipment.

The tool has been greatly enhanced by adding a descriptive XML-based language to allow the specification of the scenarios to be emulated (nodes, networks, interfaces, addresses, etc), together with the commands that each node must execute (for example, the configuration and start-up of BGP daemons). Besides, a parser has been developed to process the XML scenarios definition and automatically execute the commands needed to start the emulation using UML. A first version of VNUML will be made public during July.

VNUML tool has been written with generality in mind. It is a flexible and general emulation tool that can be used to emulate any Linux based network scenario. In fact, it is at present being used in the context of other WP4 sub-activities, for example, to test IPsec implementations or to make DNSSEC experiments. Moreover, it has been also found very useful to build permanent demos accessible by web of some the developments made in the project (see multihoming sub-activity).

IX Model

In order to study IPv6 new IX architectures based on Route Server, mainly in relation with the novel IX based address assignment, a general IX model has been defined. This model is derived from WP2 IX model C and it is illustrated in Figure 3-20.

The IX is made of Long-Haul providers giving transit services, Regional providers offering local connectivity, the IX infrastructure made of Route Server, Layer 3 Mediation Function and other services being offered by the IX, and the clients.

Three types of clients using ISP independent addresses (i.e. addresses assigned from the IX prefix) have been defined:

- Type A, which are clients directly connected to the IX using its own router and having a high control over routing policies.
- Type B, which are clients directly connected to the IX but through an IX managed router and delegating the control of routing policies to the IX administrators.
- Type C, which are traditional clients connected to ISPs but using IX assigned addresses.

The main objective of the model is to study how to offer clients using ISP independent addresses services like:

- Provider selection.
- Multi-homed for fault tolerance.
- Multi-homed for load balance.

[2] Detailed information about VNUML tool can be found in http://www.dit.upm.es/vnuml.

A set of scenarios has been defined in order to study and test how these services can be implemented. Basic scenarios for the three types of clients have been already emulated, showing that most of the services can be offered for IX connected clients (A and B). The case of C clients is different: as they are traditional customers connected to an ISP they only benefit from using ISP independent addresses, but its traffic follows the ISP routing policies.

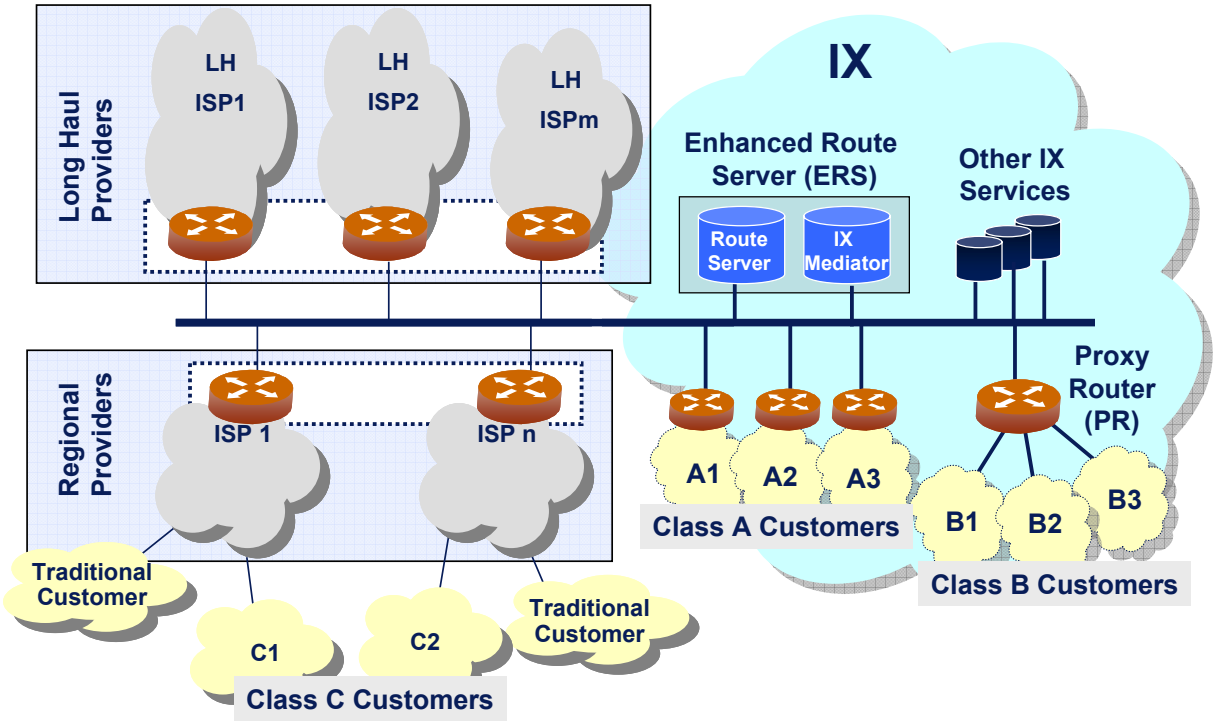

Figure 3-20: IX Model

Quagga Route Server implementation

Experimentation with quagga[3] routing daemon has continued, in particular with the BGP route server functionality. Extensive tests have been made in order to characterize its behaviour and define how it can be used to implement the IX model described above. Unfortunately, some important problems were found with the present route server implementation included in quagga.

As these problems prevented the emulation of the scenarios defined, during the second semester of Y2 we decided to develop a modified route server implementation for quagga routing daemon. This implementation will include all the functionalities needed to work over the peering model defined for the IX and shown in Figure 3-21. At present, most of the software is developed and testing is underway. Once the modifications are mature they will be contributed to quagga maintainers for inclusion in the regular code.

Finally, it is worth mentioning that the results of this sub-activity have led to the publication of an article named "Study and Emulation of IPv6 Internet Exchange based Addressing Models" in IEEE Communications Magazine special number on IPv6 published in January 2004.

[3] In principle, route server developments were foreseen over zebra routing daemon (http://www.zebra.org). However, during Y2 a new routing daemon implementation called quagga (http://www.quagga.net) was derived from zebra and found more appropriate for this sub-activity.

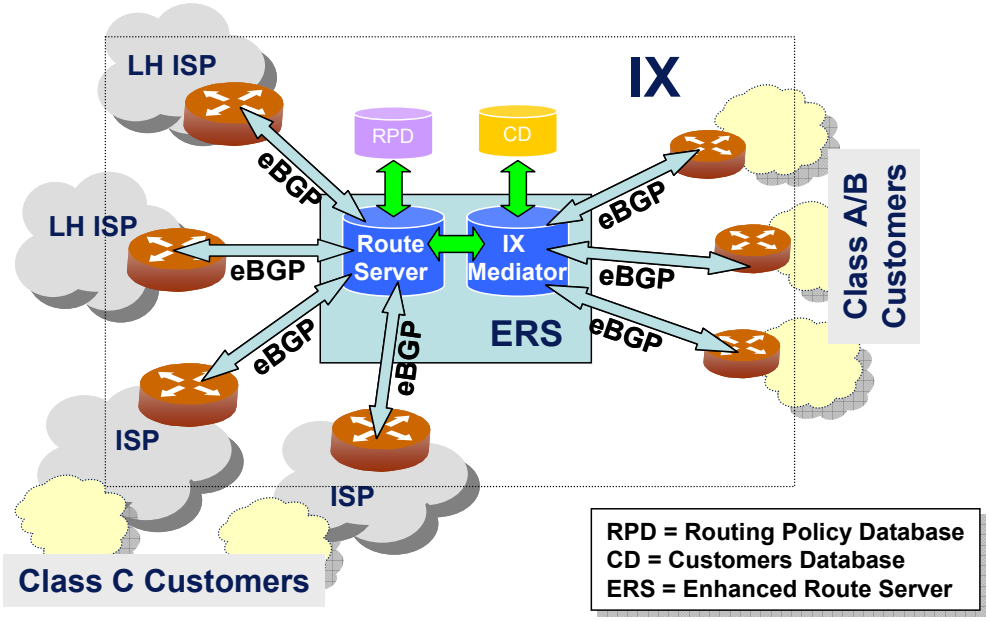

Figure 3-21: BGP peering scheme

Future Work

Work on this sub-activity will continue during Y3 on the following areas:

- Finish route server implementation in quagga routing daemon.
- Deeper study and emulation of IX based address allocation alternatives.
- Test RPSLng implementations available.
- Study Router Server functionalities within the IX for secure routing and policy management.

3.3.2 Looking Glass

Partners

UPM, TID (tester), UMU (tester)

Objectives

- Develop a basic Looking Glass application to remotely execute diagnostic tools on every partner's network (UPM).

Description

In order to test and make a diagnostics about IPv6 problems over Euro6IX network, WP3 decided on the necessity to install a *Looking Glass* software on every partners network. The basic idea was to allow other partners to test the network from one partner's point of view using typical tools like ping or traceroute, or to know how BGP routes had propagated to that partner.

Results

Starting from WP3 request, UPM developed at the beginning of Y2 a basic Looking Glass facility, partially starting from other developments found on the network. The objective was to have a simple tool adapted to Euro6IX characteristics, flexible enough to easily include new tools as needed.

The first version of the Looking Glass -a perl script written to be executed from a web server- is available at http://www.upm.euro6ix.org. It includes the following tools:

- Ping and traceroute for IPv4 and IPv6.
- MTR, a tool that combines the functionality of the 'traceroute' and 'ping' programs in a single network diagnostic tool able to estimate the network quality of the path to a destination.
- DNS, in order to query the local DNS server, allowing, for example, making recursive queries, as they are done from local machines.
- BGP and routing table queries, in order to know the status of routing at each partners' networks.

Figure 3-22 shows a screenshot of the Looking Glass software.

Apart from the normal use described above, UPM's Looking Glass scripts have been reused to show on-line demos of some services developed. For example, a demo of the multihoming developments carried out in A4.1-5 sub-activity is already available on UPM's web server.

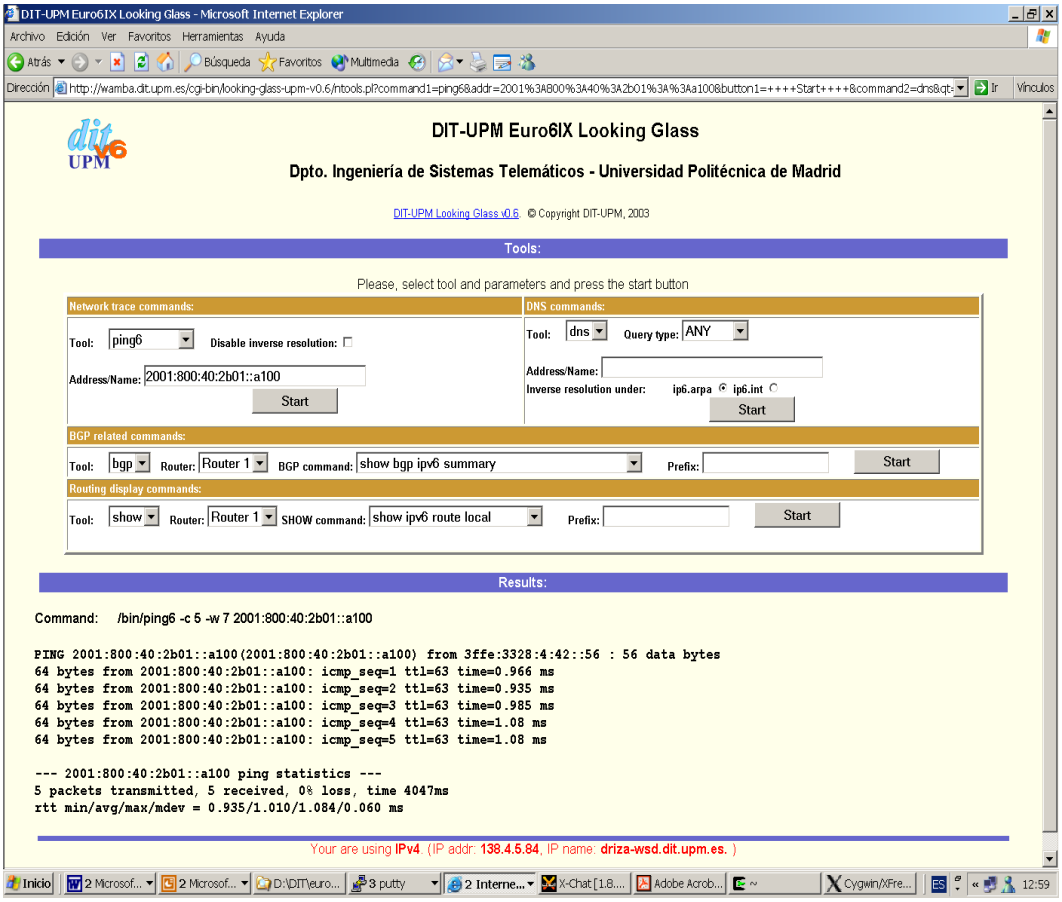

Figure 3-22: Looking Glass Interface

Future Work

No activity is foreseen for Y3, apart from the normal maintenance of the software.

3.4 Code Porting (A4.2-4)

3.4.1 Application Migration

Partners

nGn

Objectives

Code porting/Application Migration is one of the key sub-activities inside WP4. It allows programmers and finals users to run the programs widely available for IPv4 taking advantages of the new IPv6 network.

For the Y2, we have concentrated on the next goals:

- PERL
- MRTG
- PINE
- TCL

Description

PERL

Perl is one of the most widely used programming languages. Designed initially as an scripting-language for systems management, actually some developers use it as general-purpose and/or prototype-design programming language.

The current version (5.8.0) does not include the IPv6 at its core release *(stable.tar.gz)*. However, you can download separate modules to enable IPv6, i.e. Socket6. nGn contributed with a module enabling Perl with an object interface to the IPv6 API: **INET6**, implementing the same functionality as the twin INET does for IPv4.

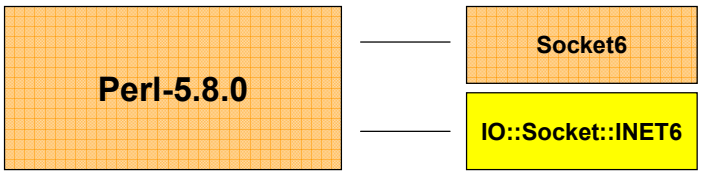

Figure 3-23: Perl and IPv6 Integration

PINE

Pine is a MUA (Mail User Agent) designed to run originally in UNIX terminal without X servers. You have management facilities, like message threading, filtering rules, address book, and access to server via IMAP or NNTP (usenet) protocol. nGn contributed to enable it on IPv6 by modifying its *network layer*, making it protocol independent.

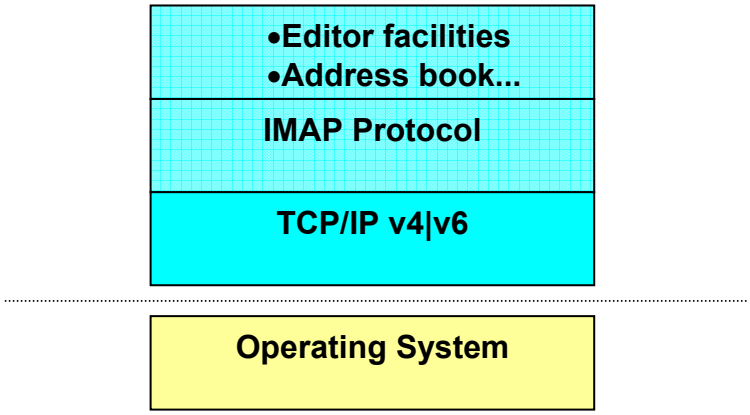

Figure 3-24: Network Layer Modified by nGn on PINE Context

TCL

TCL is powerful scripting language, executable at different platforms (Unix, win, Mac-OS) allowing some O.S facilities, file-access, socket programming…It's widely known by one of its graphical libraries, *Tcl/Tk*, allowing developers to program *X-widgets* with a relative low effort. Given the structure below, nGn modified the network component in order to let programmer to use both IPv6 and IPv4 protocols.

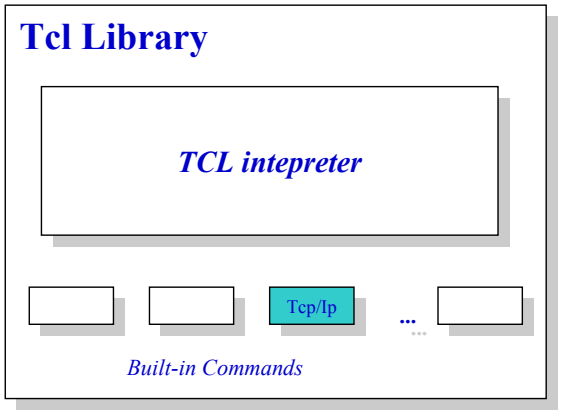

Figure 3-25: Structure of TCL Interpreter and Network Component Involved

MRTG:

MRTG is a SNMP based software tool, originally written by Tobias Oetiker, designed to retrieve data from a SNMP agent and print them in a graphic format (.png) inside HTML pages you can view further using a html browser.

Basically, it consists in a SNMP manager (client) making a request to some network device running a SNMP agent (server). Originally nGn modified the green color components, but after feedback on developers' MRTG forum only the network components were accepted from nGn.

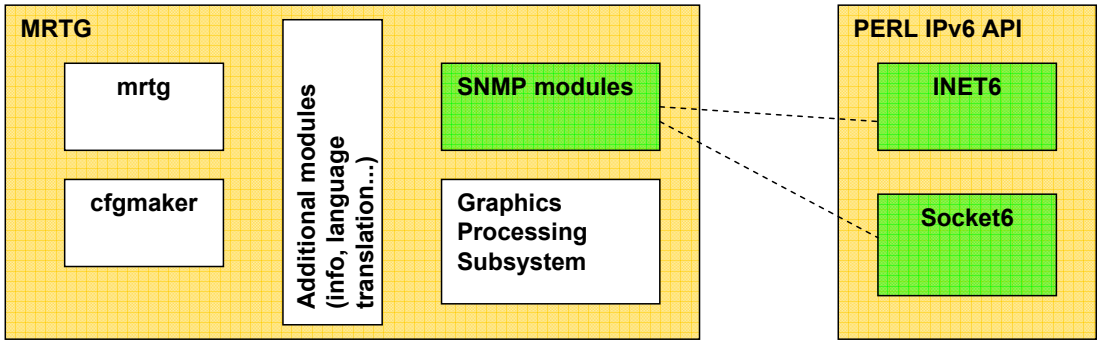

Figure 3-26: Structure of MRTG (green modules modified initially by nGn)

Relevance

Given a corporate network, MRTG would be very useful for monitoring IPv6 parameters, like incoming/outgoing average traffic, density, overload…This would support network administrators with useful statistics, evaluating the impact of the IPv6 services in order to adopt or redesign new policies.

PINE is the most widely used MUA among the UNIX community. Since the number of users for this system is increasing, we think providing one of its commonly used tools would benefit the whole community.

TCL was clearly motivated by the leading project ISABEL, according to one of the past review's issues, encouraging for synergy among activities.

Results

The results achieved during Y2 in this sub-activity are the following:

- PERL: INET-1.28 version released at http://www.cpan.org.
- MRTG: Officially Released at http://people.ee.ethz.ch/~oetiker/webtools/mrtg/pub.A first demo was run at *Madrid 2003 Global IPv6 Summit*.
- PINE: Received the *acknowledge* to include it in future versions from official vendor. Tested at least on i386-*-linux,i386-sun-solaris,i386-freebsd5.6 and hpp1.1-hp-hpux11.11 http://www.ngn.euro6ix.org/IPv6/pine.
- TCL: It has been lift to a TIP proposal (#162) at http://tcl.sourceforge.net, where it will stand for feedback, future voting and official release.

Future Work

- MRTG & PERL: MRTG project depends strongly from INET6, and this is not fully available for some platforms, i.e. *linux* (*freebsd* and *solaris* are). A last communication failure with foreign maintainers originated this bug. We are working hard to release a universal, protocol-independent INET6 platform.
- OCAML: We found some interesting *e-donkey* software written on this language. Extending OCAML to IPv6 would be the first step to migrate the application into IPv6.
- PVM: An scratch version is available, ready to run a set of benchmark programs.
- TCL: After official release, we could consider to extend the original API to deal with UDP sockets, both IPv4| IPv6. Actually only TCP sockets are programmable on IPv4.

3.4.2 Code Porting Guide

Partners

UPM, nGn

Objectives

The transition between today's IPv4 Internet and the future IPv6-based one will be a long process during which both protocol versions will coexist. One key task during this transition period is the applications porting. For that purpose, it is important to have simple and clear guidelines for developers to easily port existing applications.

The objective of this sub-activity is to continue with the work carried out inside the LONG Project in this area. Its main result was the document "Programming guidelines on transition to IPv6", which will be used as a starting point, reviewing and completing the work with special attention to show how to develop IP version-independent applications, as well as to operating systems like Windows and languages like Java not covered in the document.

Description

Existing applications have typically been written including IPv4 specific details in the source code. Only very recently IPv6 support or even IP-version independence support has been taken into account when developing applications. Although an important number of IPv6 capable basic distributed applications are already available now, there is still much work to be done.

The aim of this sub-activity is to provide general recommendations to be taken into account during the process of porting applications and services to IPv6. This will allow developers to move smoothly their applications into the new network environment.

The sub-activity is focused on how to analyze existing applications, in order to locate those parts of the code that usually should be reviewed and modified during transition to IPv6. We put special attention in most popular operating systems programming interfaces: Windows, Solaris and Linux. All of them are based on the basic socket interface defined in the classic UNIX interface. However, all of them introduce additional functionalities mainly devoted to flexibly developments when dual stack environment is in mind.

The Porting Guide includes some examples of the code porting process, used to illustrate the required changes in client and server components. Porting guidelines are valid for any programming language, however for simplicity, application examples are provided mainly in C language.

To complete the document, UPM has been working on generating programming guidelines for different programming languages, specially focusing on Java and TCL introduced by nGn. The new document release includes an appendix with detailed description of how to proceed with other programming languages like Java. The work has been mainly based on SUN JDK in a Suse Linux 8.1 distribution.

Relevance

Application transition is a key for the success of IPv6. Existing applications are written assuming IPv4. Only very recently IPv6 has been taken into account. Unless most of basic distributed applications are available now; there is too much work to do yet, because local applications and mainly the most new versions must be adapted to the new environment.

This sub-activity is devoted to provide code-porting guide to help developers to move smoothly their applications. A clear porting guide will reduce significantly the development time, but also the analysis time to search and identify where and how the code should be rewritten.

The guide has been used to integrate new applications in the Euro6IX Platform, not only in the final user side but also in the core network management environment when previous IPv4 tools should be adapted to the new IPv6 framework.

Results

During this period, the initial version has been improved with features of some programming interfaces related to Java and TCL programming languages.

The guide has been completed with description of some examples not only in C but also in Java and TCL languages. This version maintains the simple example of the DAYtime to show the porting process in previous mentioned programming languages.

Future Work

After the second project year the guide is considered finished. However, we will continue the analysis of new project developments looking for new transition problems not included in the porting guide.

4. SUMMARY AND CONCLUSIONS

This document has presented the objectives and results of all the research and innovative tasks carried out during the second year of Euro6IX project in the context of WP4 A4.1 and A4.2 activities.

As mentioned before, WP4 work has concentrated on the integration of the most important network and application activities, which were consolidated during the first year. The results from this integration work, which shall lead next year to a stable "Euro6IX Integrated Service Framework" deployed in the network, have already been shown.

Special care has been taken on the study and experimentation about how the services being tested in Euro6IX can be adapted and integrated into IXs architecture, taking advantages of being deployed in a privileged location from the topological and inter-organizational point of view.

Work around security has been very active during Y2. Starting from the isolated experiments that took place during the first year around IPsec, PKIv6 or DNSSEC, work has focused on introducing security on Euro6IX test-bed, for example, integrating PKI services into IXs, linking PKI and IPsec or starting a DNSSEC pilot service, as well as continuing the work with still immature aspects like IPv6 firewalls, secure routing protocols or IDS for IPv6.

Another example of integration results can be found in mobility and roaming evaluation sub-activity, where work has been focused on the deployment of a multidomain mobility scenario integrated in the IXs and based on MIPv6 and AAAv6. Furthermore, transition activities have been centered on two of the mechanisms that best fit into the IX architecture: NAT-PT and Tunnel Broker. Besides, investigations about how to provide IPv6 services to host located on private IPv4 networks accessible through NAT has been started.

Work around multihoming has been started during this second year, mainly with the study of the different solutions being proposed. However, a first interesting result is already available in the form of a useful enhancement to the source address selection implementation in USAGI IPv6 Linux implementation.

Besides, work has also continued on QoS, with the specification and deployment of a QoS model for Euro6IX network, and multicast, designing and deploying in cooperation with WP3 of a multicast service over Euro6IX test-bed connected to the M6Bone initiative.

In relation to IPv6 applications development, a wide and realistic range of applications is being covered in order to support the validation of the Euro6IX tested, as well as for the realization of trials. Advanced applications like multimedia, groupware or peer-to-peer are being ported and adapted to IPv6 and integrated into the Euro6IX service framework, with a special focus on security and mobility. Advanced applications are being complemented with the porting of generic software libraries and packages, as well as applications, considered important for the transition to IPv6.

Work on network management and operation tools has continued, giving rise to a mature version of Magalia distributed and inter-organization management tool, as well as a first version of Topaz IDS.

Finally, IX support tool activities has provided also interesting results, mainly in the form of a flexible tool, initially thought to emulate and study complex IX scenarios but designed to be a general tool that can be used to emulate any Linux based network scenario. In fact, it is at present being used in the context of other WP4 sub-activities, for example, to test IPsec implementations or to make DNSSEC experiments. Besides, a detailed model of an IX with address assignment has been defined in collaboration with WP2 and some of the scenarios derived from it have been already tested.

In summary, effort invested around A4.1 and A4.2 is progressing well towards the general objectives defined for WP4 inside Euro6IX project.

5. REFERENCES

The Technical Reports associated to the research sub-activities described in this document provide more detailed information about some of the results presented in it. They all can be found in the section "Documents|Deliverables" of the Euro6IX web server located at http://www.euro6ix.org.