Guillermo Cicileo

Roque Gagliano

Christian O´Flaherty

Mariela Rocha

César Olvera Morales

Jordi Palet Martínez

Álvaro Vives Martínez

IPv6 for All

A Guide for IPv6 Usage
and Application
in Different Environments

Internet Society

ISOC-AR
Chapter
Argentina

Guillermo Cicileo

Roque Gagliano

Christian O'Flaherty

Mariela Rocha

César Olvera Morales

Jordi Palet Martínez

Álvaro Vives Martínez

IPv6 for All

A Guide for IPv6 Usage and Application in Different Environments

Internet Society

ISOC-AR
Chapter
Argentina

Preface

Although the content of this book keeps its validity since the moment it was first written in 2008, some events are worth mentioning while editing its English version.

The 3rd February 2011, the IANA (Internet Assigned Numbers Authority) allocated the last blocks of IPv4 addresses to each Regional Internet Registry (RIR). Also, the 14th April 2011, the Asia Pacific Regional Internet Registry (APNIC RIR) entered the last stages of IPv4 allocation, making much difficult for organizations in that region of the world to obtain the IPv4 address space that they would justify for under previous policies. The remaining regions will follow in the near future.

These series of events reinforce the relevance of this work and the need for all kind of networks (Internet Service Providers, Enterprises or Small Businesses) to deploy IPv6.

As the international community has gained experience on the use of the new protocol, some technology trends are clearer today than at the time of our first edition, and as a consequence there has been important advances on technologies for a "world without IPv4" in some part of the network. We can mention the work on DS-Lite or 4rd as examples.

The standardization work on translation technologies between IPv4 and IPv6 is almost over and the use of the new standards (i.e. NAT64 and DNS64) is been planned in IPv6 only scenario, and it seems to be a possible path in IPv6-only cellular networks.

When we looked at the current status of IPv6 deployment, the numbers are not very encouraging. Only less than 1% of the hosts in the Internet had a working IPv6 address[*]. However, the number of BGP networks implementing IPv6 has rapidly increased in the last 2 years to around 15%[**]. We should expect this increase in the number of networks that have IPv6 available to translate on a rapid increase on IPv6 hosts count. The deployment of IPv6 is also benefiting from some industry trends, such as the replacement of Windows XP systems (without IPv6 stack enabled by default) by Windows 7 systems (with IPv6 enabled by default).

This lack of synchronization between the end of IPv4 and the global availability of IPv6 addressing at hosts and content providers will require in some parts of the world the implementation of translation, either NAT444 or NAT64. Service Providers are preparing themselves for this scenario through the deployment of "Carrier Grade NATs". The only

alternative to make sure that this change in the architecture of the Internet is temporary is the deployment of IPv6. That is the challenge that this book addresses and we are happy you are joining us in this effort.

(*) Reference: http://www.google.com/intl/en/ipv6/statistics/
(**) Reference: http://bgp.potaroo.net/v6

Acknowledgements

Contents

Authors

Guillermo Cicileo

Guillermo Cicileo is currently the General Coordinator of RIU, the network of Argentine national universities. He has been a part of the FLIP6 (Latin American IPv6 Forum) Evaluation Committe since 2007. He actively participated in the creation of CLARA (Latin American Cooperation of Advanced Networks) and was a member of the project's initial Technical Commission. He was then in charge of coordinating CLARA's Multicast Working Group from 2005 to 2008 and member of the IPv6 and Advanced Routing Working Groups. He has also participated as an instructor in the advanced routing workshops organized by CLARA, providing training on multicast, IPv6, BGP and other subjects. Prior to this, he was deputy director of the RETINA network where he was in charge of the Operations and New Technologies departments. His activities included implementing native IPv6 in the network, both its international connectivity as well as its deployment at national level. His work has been linked to national and international scientific and academic networks, areas with which he has been involved for more than 15 years. His professional background includes leading the first Argentine connection to Internet2 and Advanced Networks, as well as the country's incorporation to RedCLARA.

Roque Gagliano

Roque Gagliano has been working with IP networks for more than 12 years. He is currently a Consulting Engineer at Cisco Systems, where his responsibilities include the development of technology intelligence tools and senior management advising. Previously, he was a Senior Project Engineer and Policy Officer at LACNIC, the Internet registry for Latin America and the Caribbean. His responsibilities included coordinating technical projects and overseeing the policy development process for Internet resource allocation in the LACNIC region. He also provides IPv6 training and presentations in LACNIC in Latin America. His experience in the field of IPv6 includes designing the solution for LACNIC's corporate network in Montevideo and the critical server network in Brazil. He has also participated in the launching of the first IPv6 connections in countries or territories such as Haiti, Saint Maarten, Curaçao, and Trinidad and Tobago. Roque is also active within the IETF, particularly in the IPv6 and traffic exchange related groups. Prior to this, he was employed as a network architect at ANTEL, Uruguay, where he designed a solution to implement IPv6 over the company's MPLS network. In the past he has also worked for Sprint Nextel Corp. in the United States. Roque holds an M. Sc in Electrical Engineering from the University of Kansas, USA, and a degree in Electrical Engineering from UDELAR, Uruguay. He was awarded a Fulbright-OAS scholarship and is a member of the IEEE and ISOC.

Christian O'Flaherty

Christian O'Flaherty holds a BA in Computer Sciences awarded by the Universidad Nacional del Sur, Bahía Blanca, Argentina. He began his professional carreer teaching Networks and Operating Systems as well as Data Teleprocessing, and later went on to work in network operation and planning at the National Academic Network, RETINA. He was later Internet operations manager at Impsat Argentina, a satellite services provider that evolved into a regional IP services provider. In 2006 this company was acquired by Global Crossing, where Christian became Internet Product manager until 2009, when he assumed the position of Senior Education Manager at the Internet Society, a position he holds to this day. Between 2004 and 2008 he served as moderator of the policy mailing list and chaired LACNIC's Public Policy Forum. He is currently a member of the board of directors of ISOC Argentina and the Argentine IPv6 Task Force.

Mariela Rocha

Mariela Rocha has been involved in new technologies and network engineering since graduating as an Informations Systems Engineer from the Universidad Tecnológica Nacional, Argentina, mainly within the academic environment. She first became involved with IPv6 in 2003, participating in workshops and training activities organized by Florida International University (FIU), while working at the National Academic Network (RETINA), where she helped consolidate IPv6 deployment.

She has provided numerous trainings on IPv6 at Argentine universities, Internet Service Providers and other organizations such as CABASE (IXP of the Argentina). She has also been a frequent presenter on IPv6 in the region of Latin America.

Since 2006, she is the coordinator of the Latin American IPv6 Forum and of the Latin American and Caribbean IPv6 Task Force.

Mariela is currently technical coordinator of the Argentine University Interconnection Network, where she applies her expertise to support the deployment of new technologies.

César Olvera Morales

César Olvera Morales holds a degree in Physics awarded by the Universidad Nacional Autónoma de México (UNAM). Between 1998 and 2002 he worked at DGSCA-UNAM, where he coordinated the Interoperability Laboratory, conducted research and testing on IPv6, QoS, Multicast, MPLS, etc., organized conferences and seminars on these technologies, and participated as speaker at national and international events. In 2002 he joined Consulintel, where he participates in several IST and PROFIT projects, focusing his work on IPv6 network research, testing, design, and deployment and aspects such as routing, PLC, QoS, Multicast, MPLS, VPN, security, etc. He has cooperated with ETSI, the IPv6 Forum, Spirent, Agilent, Ixia, etc., in designing and executing interoperability, compliance and functionality testing on IPv6 devices; he has also cooperated with IETF IPv6 working groups. He has also provided IPv6 training in Latin America and Africa.

**Jordi
Palet Martínez**

Jordi Palet Martínez has worked in the field of computers, networks and telecommunications for the past 25 years and is currently CEO/CTO of Consulintel. Jordi's experience includes, among others, programming in different languages, porting operating systems, electronic circuit and microcomputer design, consultancy services, and network implementation and design. For many years he has been involved in activities organized by the IETF, ISOC, ICANN, the IPv6 Forum, IPv6 Cluster, IPv6 Task Forces and the RIRs, frequently offering training workshops on IPv6 around the world, and has authored many articles, books and documents on IPv6. He has led and/or participated in multiple research, development and innovation projects, the majority of which, such as, for example, 6SOS, Autotrans, Euro6IX, Eurov6, 6POWER, 6QM, 6LINK, ENABLE, RiNG and PlaNetS are related to IPv6. In addition to IPv6, he has worked in technologies relating to PLC/BPL, IP mobility, security and routing, among others. He is a regular presenter at IPv6-related conferences and events and part of numerous committees, including the FLIP6 (Latin American IPv6 Forum) Evaluation Committee since its creation. He cooperates closely with AfriNIC, APNIC, ARIN, and LACNIC on IPv6 dissemination and training activities.

**Álvaro
Vives Martínez**

Álvaro Vives Martínez holds a degree in Telecommunications Engineering specializing in ICT from Universidad de Vigo. After participating in a European I+D project involving digital television and the development of a DVB-MHP set-top box and working as a guest professor at Universidad de Vigo, in 2002 he went on to work at Consulintel. While working at Consulintel he has participated in several IPv6 related I+D projects both at Spanish as well as at European level: 6SOS, Euro6IX, 6POWER, 6QM, Eurov6, ENABLE, RiNG and 6DEPLOY. He has been in charge of production services (among others, DNS, http and FTP), network administration, and application development; he has tought courses and presented lectures, worked on consultancy projects in Europe, Latin America and Africa and he has carried out standardization tasks at the IETF – all of these activities related to IPv6.

1. Introduction

1.1. Current situation

As we already know, IANA pool (www.iana.net) has been depleted on February 3rd 2011, and the last blocks assigned to the Regional Internet Registries isdecreasing at a significant rate, so their depletion is rapidly approaching. In fact, the APNIC one was already exhausted on April 14th 2011.

The current protocol (Internet Protocol version 4 or IPv4) supports approximately 4 billion addresses and, due to the huge success of the Internet, it is anticipated that these will be depleted in the coming years.

It is clear that, for various reasons, many of the IPv4 addresses that have been assigned are not being utilized. For some time it was believed that by optimizing the use of IPv4 addresses, recovering unused address space, and increasing the use of Network Address Translation (NAT) technologies, the demand for IP addresses could be solved without having to adopt a new version of the Internet Protocol; indeed, there are some who still maintain this line of reasoning.

However, this idea has gradually disappeared as it has become aparent that, in the medium term, a vast number of devices will require their own IP address in order to be able to connect to the Internet and that many of these devices will even require several addresses. Even with an optimal use of IP address space, the 4 billion addresses allowed by the IPv4 protocol will not be enough.

It is important to highlight the fact that, although NAT has so far allowed the Internet to continue growing, this technology implies the loss of end-to-end connectivity and, therefore, hinders the deployment of end-to-end (client-to-client) applications and services, making the development of those services and applications more complex or costly and consequently standing in the way of innovation.

The new IPv6 protocol supports 340 trillion trillion trillion (sextillion) addresses, which makes the number of IPv4 addresses appear insignificant. Along with this larger address space, IPv6 offers a variety of advantages for network administration in terms of stability, flexibility, and simplicity. It is also likely that the *"IPv6 Era"* will generate a new wave of innovation concerning applications and services, as it will put an end to the need for shared addresses.

Gradually, IPv6 networks are being implemented and, during this transition, IPv6 and IPv4 will coexist for many years to come. Although most of the technical work having to do with the protocol has already been completed, IPv6 is yet to be deployed in Internet Service Provider networks.

1.2. Latin America and the Caribbean: On the right path

The journey towards IPv6 adoption and promotion has been slow but steady.

In 2005, for example, LACNIC –the Internet Address Registry for Latin America and the Caribbean– organized the first *"IPv6 Tour"*, which consisted of 10 events held in 10 different countries around the region. More than 3,500 people participated in these events, which were "awareness-creating" events where it was assumed that participants had no prior knowledge on the subject. Now, four years later, the situation has changed dramatically.

LACNIC, with the help of other partners of the 6DEPLOY Project – a project co-funded by the European Commission – has organized training activities in more than ten countries in which more than 800 people have participated. The major difference is the type of activities that are being organized. It is no longer necessary to explain what IPv6 is, but rather the new workshops focus on practical aspects relating to IPv6 implementation. Many participants leave these workshops already prepared to put into practice plans for the adoption of the new version of the protocol within their organizations and to request IPv6 addresses from LACNIC. This process has already been set into motion.

Today there are Internet exchange points (IXPs) in at least six Latin American and Caribbean countries that already provide services with native IPv6 in their infrastructure. Seventy-five percent of country code Top Level Domains (ccTLDs) resolve their country's DNS domain name over IPv6 through, at least, one of their primary and/or secondary servers.

During the first nine months of 2009, more than 60 Latin American and Caribbean organizations and companies received IPv6 address prefixes, which is a significant increase as compared to the 47 assignments made by LACNIC and the national registries of Mexico and Brazil during the whole of 2008.

Some operators are already providing services over IPv6 to their customers, and the list of facts and indicators that show how the transition towards IPv6 has advanced and consolidated could go on and on.

Important progress has also been made at government level. IPv6-related topics are a normal part of the agendas of organizations such as CITEL (the Inter-American Telecommunication Commission), the CTU (the Caribbean Telecommunication Union) and other government forums, where some resolutions have been approved that show governments' commitment to participating in the promotion of IPv6 and adopting the new protocol in their own infrastructure.

¿Is this enough? Clearly not; but, as mentioned earlier, these elements show that we are on the right track.

Now that the IANA IPv4 central pool is exhausted, it is clear that we must speed up thepace and move forward with greater conviction than ever towards IPv6 adoption, as what is not done soon will generate greater costs in the future.

The work carried out by organizations such as LACNIC and the Internet Society (ISOC) contribute significantly to mitigate the potential negative consequences of the transition towards IPv6. Ultimately, IPv6 is necessary for the continuity, stability, and evolution of the Internet.

The purpose of **"IPv6 for All"** is to promote the use of IPv6 in the most common environments, providing the necessary know-how and expertise so that more individuals and organizations can, in the short term, attain goals that will allow them to move this process forward. **"IPv6 for All"** contains practical configuration examples that will allow readers to experiment with the use of the new protocol following the configuration guidelines set forth in this book.

1.3. What is ISOC?

The Internet Society (ISOC) is an independent international, non-for-profit organization with offices in Geneva, Switzerland, and Reston, Virginia, USA. ISOC acts as a center for the global exchange of technically reliable and objective information about the Internet, as a provider of education, and as a facilitator and coordinator of Internet-related initiaties around the world. It is the organizational home for the IETF (Internet Engineering Task Force), the IAB (Internet Architecture Board) and the IRTF (Internet Research Task Force).

ISOC was founded in 1992 to provide leadership in Internet related standards, education, and policy. It is backed by an active international network of members that help promote and achieve ISOC's mission within the entire Internet community and around the world. The Internet Society has more than 80 organizational and more than 28,000 individual members in over 80 chapters around the world, all of which contribute to regionalize the scope of ISOC's technology, education and policy initiatives.

The Internet Society's website can be found at http://www.isoc.org.

1.4. Internet Society Argentina Chapter

Internet Society Chapters are groups of individuals who, either because they live in a particular geographical region (for example a particular city or country) or because they share a specific interest on Internet related issues, voluntarily organize themselves

and decide to carry out, as ISOC members, different activities that are in line with the organization´s goals and principles.

The ISOC Argentina Chapter (ISOC-Ar) is an independent, democratic, nonprofit civil organization that operates within the framework of the Asociación Civil de Argentinos en Internet. Founded in 1999, ISOC-Ar was recognized as a legal entity by Resolución IGJ N° 297/2000. Its goals include promoting the open development and evolution of the Internet, its services and contents for the benefit of all, particularly of the residents of the Argentine Republic, by promoting global Internet community activities that will foster close communication and bring together the members of the Internet Society who live in the country.

In line with the Internet Society's mission and goals, ISOC-Ar members permanently conduct different activities aimed at supporting the Guiding Principles. Thus, for example, since 2007 we have held the Accessibility Seminars: "Towards an Internet without barriers for persons with disabilities", where issues concerning the difficulties that people with disabilities must face in order to use the Internet are discussed, as are the best practices in force that seek to reduce this divide. Other examples include participation at different events and seminars, co-organizing the Usability Day held in November 2008, and the execution of projects funded by the Internet Society.

1.4.1. Community Grants Program

The Community Grants Program is one of the activities promoted by ISOC for its members and Chapters.

The goal of these programs is to provide financial support that will allow the execution of projects that serve to:
- Advance ISOC's mission and goals, specifically those aligned with ISOC Major Strategic Initiatives and Principles;
- Serve the Chapters' communities;
- Nurture collaborative work among Chapters or individual members;
- Enhance and utilize knowledge sharing in the global Internet community; and
- Encourage Chapters' sustainability and relevance.

1.4.2 The "IPv6 for All" Project

This book was created with the aim of providing the Internet community, both local as well as global, with a manual that includes the tools necessary to encourage and promote the adoption of the IPv6 protocol in different environments, and was also motivated by the concerns that have arisen regarding the new protocol's late adoption.

Structured in Chapters each corresponding to a specific environment, **"IPv6 for All"**

explains, in a clear and simple manner, the steps and requirements involved in configuring and implementing the new version of the IP protocol in environments as diverse as Residential Networks, Research and Education Networks, Companies, Internet Service Providers (ISPs), End Users, or Services.

This Project, on which ISOC-Ar had been working for several years, has been made possible thanks to the financial support of the Internet Society. The book´s content was created with the cooperation of both local and international experts who shared their knowledge and expertise, thus contributing towards this slow but inexorable journey towards IPv6 adoption.

Mónica Abalo Laforgia
President of the Internet Society Argentina Chapter

Sebastián Bellagamba
Manager of the Internet Society Regional Bureau for
Latin America and the Caribbean

Raúl Echeberría
Executive Director of LACNIC
Member of the Internet Society's Board of Trustees

2. End Users

1. Introduction

This chapter contains an introduction to the setup and basic configuration of IPv6 in different end-user platforms (operating systems).

The following operating systems are considered:
- Windows XP
- Windows Server 2003
- Windows Vista
- Windows Server 2008
- Windows 7
- Windows 2000
- Mac OS X
- Linux
- BSD

Please note that, due to the large number of versions that exist in some cases, particularly in the case of Linux and BSD, this book presents generic examples; therefore, depending on the particular version of the operating system in use, slight differences may exist which the reader must sort out with the help of the operating system's documentation.

2. Setting up IPv6

Most operating systems have included some type of IPv6 support since 2001.

It's true that, initially, in some cases, IPv6 was not officially supported by the manufacturers but instead they included test versions of the IPv6 protocol stack.

This was the case of IPv6 support in Windows 2000 (and in earlier versions of Windows NT which, due to their age will not be described in this document), and also in the first version of Windows XP before what is known as Service Pack 1 (SP1) was launched.

It is becoming increasingly common for different platforms or operating systems to not only incorporate IPv6 but to have it factory enabled by the manufacturer; therefore they do not require any further user intervention.

The above is true not only for desktop or laptop computers but also for other devices that use the same operating systems or reduced versions of these, such as mobile telephones, PDAs, gaming platforms, etc. Logically, in some cases those reduced versions of the operating systems don't include all of the original operating system's features and, therefore, it may not be possible to access all the features described for IPv6 configuration and testing.

2.1 Setting up IPv6 on Windows

The most recent Windows platforms include one of the most comprehensive IPv6 stacks available:
- Windows XP SP1 and later
- Windows Server 2003
- Windows Vista
- Windows Server 2008
- Windows 7

As mentioned earlier, some Windows platforms were initially developed as technology previews and therefore have more limited features and lack manufacturer support:
- Windows XP without SP1
- Windows 2000 up to and including SP1

A Windows NT 4.0 version for developers also exists, but this document will not cover this operating system in detail.

Finally, third-party products –without Microsoft support– have been developed for for:
- Windows 95/98/ME
- Windows 2000 with SP2 and later

In general, supported functionalities include those listed below, though some may only be available in the most recent versions of each software:
- Autoconfiguration
- 6in4 tunneling
- 6to4 tunneling
- 6to4 relay
- Teredo tunneling
- ISATAP tunneling
- IPsec (manually configured keys)

2.1.1 Setting up IPv6 on Windows XP/2003

It could in fact be said that IPv6 is already installed both on Windows XP as well as on Windows Server 2003 and, therefore, rather than speaking of setting up IPv6 we should speak of enabling the protocol.

Two procedures exist for enabling IPv6 on these two platforms:

2.1.1.1 Command line

In a DOS window, execute the command: **ipv6 install**
After a few seconds a confirmation message will notify us that the installation has been successful.

Depending on the version of the operating system**,** the following command could also be used: **netsh interface ipv6 instal**l

2.1.1.2 Graphical user interface

In the graphical environment or control panel, go to "Network Connections", select "Local Area Network" or "Wireless Network", right-click on "Properties" and click on "install", "protocol" and select "Microsoft TCP/IP version 6".

The result will be similar to the one shown in the following screenshot:
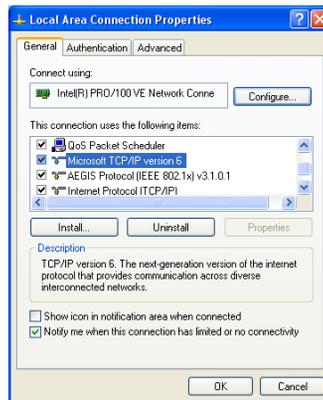
Figure 1: **Screenshot showing IPv6 installation on Windows XP/2003**

2.1.2. Setting up IPv6 on Windows Vista/2008

Windows Vista has always had IPv6 support already installed and enabled by default, therefore no additional configuration is required. If for some reason it has been disabled, the netsh or graphical user interface procedure described for Windows XP/2003 could be used to re-enable it.

Bear in mind that netsh requires a DOS window with administrator privileges.

In Windows Vista IPv6 has additional features as compared to Windows XP/2003, such as, por example:
- Full IPsec support
- MLDv2
- Link-Local Multicast Name Resolution (LLMNR)
    - Does not require a DNS server IPv6 nodes in a network segment request their names from a multicast IPv6 address. Similar to NetBIOS operation.
- Supports IPv6 addresses in URLs
- IPv6 Control Protocol (IPV6CP - RFC5072)
- IPv6 over PPP
- DHCPv6 as client and server

End Users

- Random interface identifier by default (RFC3041)
- Teredo supports symmetric NATs
  - Enabled by default. It is only used if an application requires IPv6 support and native IPv6 connectivity is not available.

The user can check that it is installed via the command line or the graphical user interface, in a manner similar to that described for Windows XP.
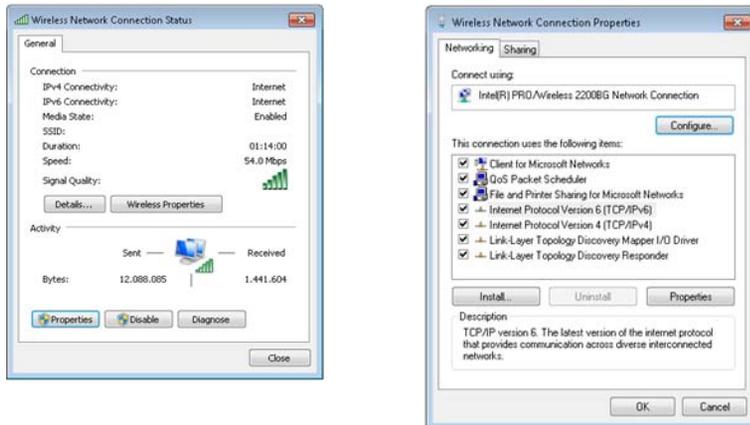
FIGURE 2: **NETWORK CONNECTION PROPERTIES AND IPV6 INSTALLATION ON WINDOWS VISTA**

2.1.3. Setting up IPv6 on Windows 7

Just as in the case of Windows Vista/2008, IPv6 is already installed and enabled by default on Windows 7. Likewise, if for some reason it has been disabled, the netsh or graphical user interface procedures described for Windows XP/2003 could be used to re-enable it.

Bear in mind that netsh requires a DOS window with administrator privileges.

The following is a summary of the features included in this version:
- IPv6 support similar to that of Windows Vista and Server 2008
  - IPsec, MLDv2, LLMNR, IPv6 in URLs, IPV6CP, IPv6 over PPP, DHCPv6, Teredo
  - Changes: Random interface identifier by default (RFC3041)
    - Does not use EUI-64 by default for the interface identifier in autoconfigured addresses.

- New enhancements:
  - IP-HTTPS (IP over Secure HTTP)
    - Allows hosts to traverse a proxy server or firewall and connect to private networks via IPv6 inside an HTTPS tunnel. HTTPS does not provide data security; in order to provide security to an IP-HTTPS connection IPsec must be used. More information is available at *http://msdn.microsoft.com/en-us/library/dd358571.aspx*

IPv6 for All. A Guide for IPv6 Usage and Application.

- DirectAccess
  - Allows users to connect seamlessly to the corporate network without specifically establishing a VPN connection. It also allows the network administrator to continue in contact with the mobile hosts outside the office, as well as perform updates and provide support to those devices. In this network architecture an IPv6 client communicates with an IPv6 server on the corporate network. It is possible to connect to the Internet via an IPv4 connection using 6to4, Teredo and ISATAP tunneling. IP-HTTPS can also be used. DirectAccess uses IPsec tunneling to provide security to resource access and authentication.
  - The client can be Windows 7 or Server 2008. The server can be Windows Server 2008.

Just as in the case of Windows Vista, its set-up may be verified using the graphical user interface:
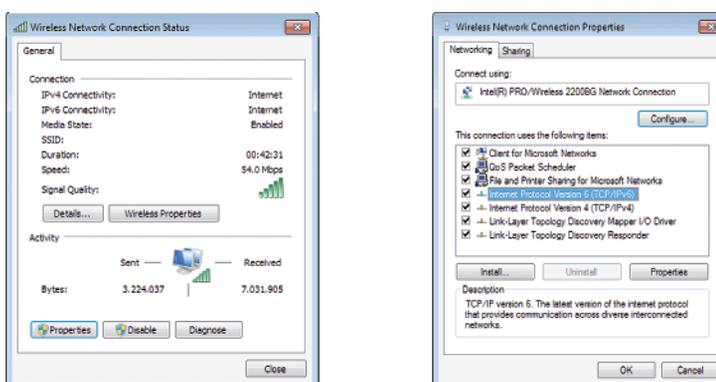
Figure 3: **Network connection properties and IPv6 installation on Windows 7**

2.1.4. Setting up IPv6 on Windows 2000

Installing the IPv6 stack for Windows 2000 requires downloading the code corresponding to the IPv6 stack as, unlike the operating systems mentioned so far, IPv6 is not preinstalled by the manufacturer.

As mentioned earlier, Microsoft does not officially support IPv6 on Windows 2000, as it was released as a technology preview version.

For this reason, we must first download Microsoft IPv6 Technology Preview for Windows 2000:
- tpipv6-001205-SP2-IE6 for SP1 and SP2
- tpipv6-001205-SP3-IE6 for SP3
- tpipv6-001205-SP4-IE6 for SP4

All of these files are available at:

http://www.sixxs.net/faq/connectivity/?faq=ossetup&os=windows

After the download is complete, the installation procedure is as follows:

- Login to the system with local administrator privileges

- Extract the IPv6 Technology Preview files, for example, to C:\IPv6Kit

- Follow the SPn & IE6 fixed.txt procedure to modify the /setup/hotfix.ini file

- Run Setup.exe or hotfix.exe

- On the Windows 2000 desktop, click on Start, then Settings, then Network Connections. Alternately, right-click on My Network Places and then click on Properties.

- Right-click on Ethernet based connections for which you wish to add the IPv6 protocol and then click on Properties Usually this connection is called Local Area Connection.

- Click on Install

- In the Select Network Component Type dialogue box, click on Protocol then click on Add.

- In the Select Network Protocol dialogue box, click on Microsoft IPv6 Protocol and then click on Accept.

- Click on Close to close the Local Area Connection Properties dialogue box.

2.2. Setting up IPv6 on Mac OS X

Apple has supported IPv6 since Mac OS X version 10.2 (Jaguar) and the protocol is enabled by default. This means that no additional configuration is required.

2.3. Setting up IPv6 on Linux

Linux supports IPv6 since version 2.4x of the kernel.

To check whether it is installed:
#test -f /proc/net/if_inet6 && echo "Running kernel supports IPv6"

To install the IPv6 module:
#modprobe ipv6

The module can be checked as follows:
#lsmod |grep -w 'ipv6' && echo "IPv6 module loaded"

The automatic loading/unloading of the module can also be configured (/etc/modules. conf o /etc/conf.modules):

alias net-pf-10 ipv6 #enables loading on demand
alias net-pf-10 off #disables loading on demand

This can be permanently configured depending on the Linux version in use.

2.3.1 Permanent configuration on Red Hat (from v7.1) and similar

Add to /etc/sysconfig/network:
NETWORKING_IPV6=yes

Restart the network:
 # service network restart
Or
#/etc/init.d/network restart

2.3.2. Permanent configuration on SuSE

Add to /etc/sysconfig/network/ifcfg-<Interface-Name>:
SUSE 8.0: IP6ADDR="<ipv6-address>/<prefix>"
SUSE 8.1: IPADDR="<ipv6-address>/<prefix>"

2.3.3. Permanent configuration on DEBIAN

Once the IPv6 module is loaded, edit /etc/network/interfaces, for example:
iface eth0 inet6 static
   pre-up modprobe ipv6
   address 2001:DB8:1234:5: :1:1
   # Completely eliminates autoconfiguration:
   # up echo 0 > /proc/sys/net/ipv6/conf/all/autoconf netmask 64
   # The router is autoconfigured and does not have a fixed address.
   # It is found using
   # (/proc/sys/net/ipv6/conf/all/accept_ra).
   # Otherwise the gateway must be configured:
   # gateway 2001:DB8:1234:5: :1

Restart the network:
   # ifup --force eth0

2.4. Setting up IPv6 on BSD

BSD includes IPv6 support since version 4.5.

IPv6 support in BSD is excellent and the stack comes preinstalled, so no additional steps are required.

3. IPv6 Verification

After setting up IPv6, depending on the different platforms, there are one or more options for checking that the setup has been successful and whether or not there is connectivity both within the local network as well as with other IPv6 networks.

End Users

3.1. Verification in Windows

In addition to viewing whether the IPv6 stack has been installed using the graphical user interface, as mentioned in the setup section, we can also use the ipconfig or ipv6 if command (unavailable in the most recent Windows versions).

The ipconfig command provides IPv6 and IPv4 configuration information for the different interfaces, whereas ipv6 if only shows IPv6 related information.

For example, assuming that our Ethernet interface has index number 5 (a number that depends on each device's specific hardware), the result of **ipv6 if 5** would be as follows:

```
Interface 5: Ethernet: Local Area Connection
    Guid {F5149413-6E54-4FDA-87BD-24067735E363}
    uses Neighbor Discovery
    uses Router Discovery
    link-layer address: 00-01-4a-18-26-c7
    preferred global 2001:db8: :fde7:a76f:62d5:3bb9, life 6d21h3m20s/21h33s
      (temporary)
    preferred global 2001:db8: :201:4aff:fe18:26c7, life 9d23h51m39s/6d23h51m39s
      (public)
    preferred link-local fe80: :201:4aff:fe18:26c7, life infinite
    multicast interface-local ff01: :1, 1 refs, not reportable
    multicast link-local ff02: :1, 1 refs, not reportable
    multicast link-local ff02: :1:ff18:26c7, 2 refs, last reporter
    multicast link-local ff02: :1:ffd5:3bb9, 1 refs, last reporter
    multicast link-local ff02: :1:ff00:4, 1 refs, last reporter
    multicast link-local ff02: :1:ff00:2, 1 refs, last reporter
    link MTU 1500 (true link MTU 1500)
    current hop limit 64
    reachable time 29000ms (base 30000ms)
    retransmission interval 1000ms
    DAD transmits 1
    default site prefix length 48
```

The ipconfig command results would be as follows:

```
Windows IP Configuration

    Ethernet adapter Public Network:
        Connection-specific DNS Suffix:
        IP Address. . . . . . . . . . . . . . . . . . . . . . : 10.10.10.250
        Subnet Mask . . . . . . . . . . . . . . . . . . . : 255.255.255.0
        IP Address. . . . . . . . . . . . . . . . . . . . . . : 2a01:48:20:0:200:1cff:feb5:c535
```

```
    IP Address. . . . . . . . . . . . . . . . . . . . . . : fe80: :200:1cff:feb5:c535%4
    Default Gateway  . . . . . . . . . . . . . . . : 10.10.10.1


Tunnel adapter Consulintel:
    Connection-specific DNS Suffix:
    IP Address. . . . . . . . . . . . . . . . . . . . . . : 2a01:48:20:0:200:1cff:feb5:c535
    IP Address. . . . . . . . . . . . . . . . . . . . .  : fe80: :5:a0a:afa%5
    Default Gateway  . . . . . . . . . . . . . . . : 2a01:48:20: :d5ac:227d


Tunnel adapter Automatic Tunneling Pseudo-Interface:
    Connection-specific DNS Suffix:
    IP Address. . . . . . . . . . . . . . . . . . . . . . : fe80: :5efe:10.10.10.250%2
    Default Gateway  . . . . . . . . . . . . . . . :
```

Likewise, if the **ipconfig /all** command is used:

```
Windows IP Configuration
    Host Name   . . . . . . . . . . . . . . . . . . . . : dns1
    Primary DNS Suffix  . . . . . . . . . . . . . : consulintel.com
    Node Type . . . . . . . . . . . . . . . . . . . . . : broadcast
    IP Routing Enabled . . . . . . . . . . . . . : Yes
    WINS Proxy Enabled . . . . . . . . . . . . : Yes
    DNS Suffix Search List  . . . . . . . . . . : consulintel.com


Ethernet adapter Public Network:
    Connection-specific DNS Suffix:
    Description. . . . . . . . . . . . . . . . . . . . . . : Intel based PCI Fast Ethernet Adapter (Generic)
    Physical Address  . . . . . . . . . . . . . . . : 00-00-1C-B5-C5-35
    DHCP enabled. . . . . . . . . . . . . . . . . : No
    IP Address. . . . . . . . . . . . . . . . . . . . . . : 10.10.10.250
    Subnet Mask . . . . . . . . . . . . . . . . . . . : 255.255.255.0
    IP Address. . . . . . . . . . . . . . . . . . . . . . : 2a01:48:20:0:200:1cff:feb5:c535
    IP Address. . . . . . . . . . . . . . . . . . . . .  : fe80: :200:1cff:feb5:c535%4
    Default Gateway  . . . . . . . . . . . . . . . : 10.10.10.1
    DNS Servers . . . . . . . . . . . . . . . . . . . : 80.58.0.33
                                                         80.58.32.97
                                                         10.10.10.250
                                                         fec0:0:0:ffff: :1%1
                                                         fec0:0:0:ffff: :2%1
                                                         fec0:0:0:ffff: :3%1
Tunnel adapter Consulintel:
    Connection-specific DNS Suffix:
    Description  . . . . . . . . . . . . . . . . . . . . : Configured Tunnel Interface
    Physical Address  . . . . . . . . . . . . . . . : 0A-0A-0A-FA
    DHCP enabled. . . . . . . . . . . . . . . . . : No
```

```
IP Address. . . . . . . . . . . . . . . . . . . . . . : 2a01:48:20:0:200:1cff:feb5:c535
IP Address. . . . . . . . . . . . . . . . . . . . . . : fe80: :5:a0a:afa%5
Default Gateway  . . . . . . . . . . . . . . . : 2a01:48:20: :d5ac:227d
DNS Servers . . . . . . . . . . . . . . . . . . . . : fec0:0:0:ffff: :1%2
                                        fec0:0:0:ffff: :2%2
                                        fec0:0:0:ffff: :3%2
NetBios over TCPIP . . . . . . . . . . . . . : Disabled
```

Tunnel adapter Automatic Tunneling Pseudo-Interface:

```
Connection-specific DNS Suffix:
Description  . . . . . . . . . . . . . . . . . . . . : Automatic Tunneling Pseudo-Interface
Physical Address  . . . . . . . . . . . . . . . : 0A-0A-0A-FA
DHCP enabled. . . . . . . . . . . . . . . . . : No
IP Address. . . . . . . . . . . . . . . . . . . . . . : fe80: :5efe:10.10.10.250%2
Default Gateway  . . . . . . . . . . . . . . . :
DNS Servers . . . . . . . . . . . . . . . . . . . . : fec0:0:0:ffff: :1%1
                                        fec0:0:0:ffff: :2%1
                                        fec0:0:0:ffff: :3%1
NetBios over TCPIP . . . . . . . . . . . . . : Disabled
```

An additional test should be conducted to check that the interface is reachable using the **ping/ping6** command (depending on the specific version of each operating system, one or both of these commands may be available). Example using the loopback address:

```
ping : :1
     Pinging : :1 from : :1 with 32 bytes of data:
     Reply from : :1: time<1m
     Reply from : :1: time<1m
     Reply from : :1: time<1m
     Reply from : :1: time<1m
     Ping statistics for : :1:
       Packets: Sent = 4, Received = 4, Lost = 0     (0% loss),
     Approximate round trip times in milli-seconds:
       Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

You can also ping the own link-local address (local link, i.e. valid only in the local network segment to which the interface is connected) of a specific network adapter (the link-local address can be seen using ipv6 if or ipconfig):

```
ping6 fe80: :e8a7:b568:a076:6ba3 (own link-local)
Pinging fe80: :e8a7:b568:a076:6ba3 from fe80: :e8a7:b568:a076:6ba3%5 with
     32 bytes of data:
     Reply from fe80: :e8a7:b568:a076:6ba3: time<1m
     Reply from fe80: :e8a7:b568:a076:6ba3: time<1m
```

Reply from fe80: :e8a7:b568:a076:6ba3: time<1m
Reply from fe80: :e8a7:b568:a076:6ba3: time<1m
Ping statistics for fe80: :e8a7:b568:a076:6ba3:
   Packets: Sent = 4, Received = 4, Lost = 0   (0% loss),
Approximate round trip times in milli-seconds:
   Minimum = 0ms, Maximum = 0ms, Average = 0ms

The next step is to check connectivity with the local network. This is only possible if there are other devices with properly configured IPv6 on the local network (and if firewall settings allow using the ping command). In this case the ping command would be used as described in the previous example, except that it would be run using the link-local address (or a global address, if any) of the computer you wish to ping.

ping fe80 : : 200:87ff:fe28:a0e0%5 (neighbor's link-local on interface 5)
Pinging fe80 : : 200:87ff:fe28:a0e0%5 from fe80: :201:4aff:fe18:26c7%5 with 32 bytes of data:
   Reply from fe80: :200:87ff:fe28:a0e0%5: time<1ms
   Reply from fe80: :200:87ff:fe28:a0e0%5: time<1ms
   Reply from fe80: :200:87ff:fe28:a0e0%5: time<1ms
   Reply from fe80: :200:87ff:fe28:a0e0%5: time<1ms
   Ping statistics for fe80: :200:87ff:fe28:a0e0%5:
     Packets: Sent = 4, Received = 4, Lost = 0   (0% loss),
   Approximate round trip times in milli-seconds:
     Minimum = 0ms, Maximum = 0ms, Average = 0ms

Likewise, if there is connectivity between the local network and external ones networks, i.e. with other IPv6 devices on the Internet, results similar to the following can be obtained:

ping www.ipv6tf.org
Pinging www.ipv6tf.org [2a01:48:1:0:2e0:81ff:fe05:4658] from
2001:db8:0:0:2c0:26ff:fea0:a341 with 32 bytes of data:
   Reply from 2a01:48:1:0:2e0:81ff:fe05:4658: time=99.661m
   Reply from 2a01:48:1:0:2e0:81ff:fe05:4658: time<106.572m
   Reply from 2a01:48:1:0:2e0:81ff:fe05:4658: time<88.624m
   Reply from 2a01:48:1:0:2e0:81ff:fe05:4658: time<76.629m
   Ping statistics for 2a01:48:1:0:2e0:81ff:fe05:4658:
     Packets: Sent = 4, Received = 4, Lost = 0
(0% loss),
   Approximate round trip times in milli-seconds:
     Minimum = 76.629ms, Maximum = 106.572ms, Average = 92.871ms

An additional step would be to use a tool that shows the hops between the different points of the network, from our own computer to the destination computer, known as a traceroute. In order to do this we need to use the tracert or tracert6 command (depending on the version/platform):

tracert www.lacnic.net

Tracing route to lacnic.net [2001:13c7:7002:4000: :10]

over a maximum of 30 hops:

```
1  <1 ms   <1 ms   <1 ms  2a01:48:1: :ff0
2  29 ms   25 ms    7 ms  2a01:48: :d5ac:227d
3  53 ms   60 ms   35 ms  tunnel105.tserv17.lon1.ipv6.he.net [2001:470:14:69: :1]
4  75 ms  109 ms   34 ms  gige-g4-18.core1.lon1.he.net [2001:470:0:a3: :1]
5  63 ms   43 ms   73 ms  10gigabitethernet1-1.core1.ams1.he.net [2001:470:0:3f: :2]
6  447 ms  163 ms  112 ms  2001:7f8:1: :a500:3549:2
7  297 ms  325 ms  319 ms  2001:450:2002:7f: :2
8  303 ms  313 ms  656 ms  ar01.bb2.registro.br [2001:12ff:2:1: :244]
9  297 ms  315 ms  313 ms  gw01.lacnic.registro.br [2001:12ff:1:3: :212]
10 302 ms  320 ms  320 ms  www.lacnic.net [2001:13c7:7002:4000: :10]
```

Trace complete.

3.2. Verification in Mac OS X

The following screen can be reached through System Preferences > Network > Advanced. In the TCP/IP tab you can verify that IPv6 is automatically configured.
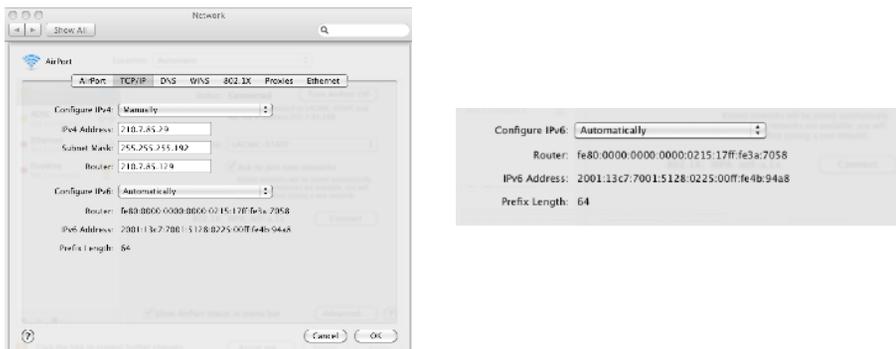
FIGURE 4: VERIFYING AUTOMATIC IPv6 CONFIGURATION IN MAC OS X

If you wish, you can also use the command line to execute the ifconfig command:
$ ifconfig

```
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
     inet6 fe80: :1%lo0 prefixlen 64 scopeid 0x1
     inet 127.0.0.1 netmask 0xff000000
     inet6 : :1 prefixlen 128
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
stf0: flags=1<UP> mtu 1280
     inet6 2002:8281:57f9:1: :1 prefixlen 16
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
     ether 00:1b:63:bd:71:67
```

media: autoselect status: inactive

supported media: autoselect 10baseT/UTP <half-duplex> 10baseT/UTP <full-duplex> 10baseT/UTP <full-duplex,hw-loopback> 10baseT/UTP <full-duplex,flow-control> 100baseTX <half-duplex> 100baseTX <full-duplex> 100baseTX <full-duplex,hw-loopback> 100baseTX <full-duplex,flow-control> 1000baseT <full-duplex> 1000baseT <full-duplex,hw-loopback> 1000baseT <full-duplex,flow-control> none

fw0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> mtu 4078

lladdr 00:1e:52:ff:fe:46:46:0c

media: autoselect <full-duplex> status: inactive

supported media: autoselect <full-duplex>

en1: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500

inet6 fe80: :21e:52ff:fe73:c2a6%en1 prefixlen 64 scopeid 0x6

inet6 2001:df8: :80:21e:52ff:fe73:c2a6 prefixlen 64 autoconf

inet 130.129.87.249 netmask 0xfffff800 broadcast 130.129.87.255

ether 00:1e:52:73:c2:a6

media: autoselect status: active

supported media: autoselect

en5: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500

ether 00:1e:52:d7:90:f5

media: autoselect status: inactive

supported media: none autoselect 10baseT/UTP <half-duplex>

en2:   flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500

inet6 fe80: :21c:42ff:fe00:0%en2 prefixlen 64 scopeid 0x8

inet 10.37.129.3 netmask 0xffffff00 broadcast 10.37.129.255

ether 00:1c:42:00:00:00

media: autoselect status: active

supported media: autoselect

en3:   flags=8963<UP,BROADCAST,SMART,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500

inet6 fe80: :21c:42ff:fe00:1%en3 prefixlen 64 scopeid 0x9

inet 10.211.55.8 netmask 0xffffff00 broadcast 10.211.55.255

ether 00:1c:42:00:00:01

media: autoselect status: active

supported media: autoselect

tun0: flags=88d1<UP,POINTOPOINT,RUNNING,NOARP,SIMPLEX,MULTICAST> mtu 1500

open (pid 199)

Just as in the case of Windows, the ping6 and traceroute6 commands may be used from a terminal window (note that in this case the commands must be written in full):

$ ping6 www.ipv6tf.org
PING6(56=40+8+8 bytes) 2001:df8::80:21e:52ff:fe73:c2a6 --> 2a01:48:1::2e0:81ff:fe05:4658
16 bytes from 2a01:48:1: :2e0:81ff:fe05:4658, icmp_seq=0 hlim=49 time=643.332 ms

16 bytes from 2a01:48:1: :2e0:81ff:fe05:4658, icmp_seq=1 hlim=49 time=87.239 ms
16 bytes from 2a01:48:1: :2e0:81ff:fe05:4658, icmp_seq=3 hlim=49 time=82.984 ms
16 bytes from 2a01:48:1: :2e0:81ff:fe05:4658, icmp_seq=4 hlim=49 time=202.559 ms
^C
--- www.ipv6tf.org ping6 statistics ---
5 packets transmitted, 4 packets received, 20% packet loss
round-trip min/avg/max = 82.984/254.029/643.332 ms

$ ping6 fe80: :21e:52ff:fe73:c2a6%en1
PING6(56=40+8+8bytes)fe80::21e:52ff:fe73:c2a6%en1-->fe80::21e:52ff:fe73:c2a6%en1
16 bytes from fe80: :21e:52ff:fe73:c2a6%en1, icmp_seq=0 hlim=64 time=0.089 ms
16 bytes from fe80: :21e:52ff:fe73:c2a6%en1, icmp_seq=1 hlim=64 time=0.117 ms
16 bytes from fe80: :21e:52ff:fe73:c2a6%en1, icmp_seq=2 hlim=64 time=0.118 ms
16 bytes from fe80: :21e:52ff:fe73:c2a6%en1, icmp_seq=3 hlim=64 time=0.167 ms
^C
--- fe80: :21e:52ff:fe73:c2a6%en1 ping6 statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.089/0.123/0.167 ms

$ ping6 www.ipv6tf.org
PING6(56=40+8+8bytes)2002:4e40:58c0:9:21e:52ff:fe73:c2a6-->2a01:48:1::2e0:81ff:fe05:4658
16 bytes from 2a01:48:1: :2e0:81ff:fe05:4658, icmp_seq=0 hlim=60 time=93.848 ms
16 bytes from 2a01:48:1: :2e0:81ff:fe05:4658, icmp_seq=1 hlim=60 time=93.32 ms
16 bytes from 2a01:48:1: :2e0:81ff:fe05:4658, icmp_seq=2 hlim=60 time=92.087 ms
16 bytes from 2a01:48:1: :2e0:81ff:fe05:4658, icmp_seq=3 hlim=60 time=89.836 ms
^C
--- www.ipv6tf.org ping6 statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 89.836/92.273/93.848 ms

Using the traceroute6 command:

$ traceroute6 www.ipv6tf.org
traceroute6 to www.ipv6tf.org (2a01:48:1: :2e0:81ff:fe05:4658) from 2001:df8: :80:21e:52ff:fe73:c2a6, 30 hops max, 12 byte packets
        1 2001:df8:0:80: :3 433.216 ms 0.813 ms 1.108 ms
        2 htg0-ncore-2.gigabiteth5-2.swip.net 1.281 ms 1.141 ms 1.072 ms
        3 avk-core-1.gigabiteth6-0-0.swip.net 1.514 ms 1.432 ms 2.269 ms
        4 avk-core-2.tengigabiteth2-1.swip.net 1.444 ms 1.476 ms 1.275 ms
        5 ibr01-tu15.stkh01.occaid.net 3.865 ms 2.842 ms 2.926 ms
        6 bbr01-p2-0.lndn01.occaid.net 43.132 ms 42.645 ms 43.049 ms
        7 neosky-ic-8241-lon.customer.occaid.net 66.522 ms 66.901 ms 67.478 ms
        8 consulintel-neosky.consulintel.es 99.245 ms 106.983 ms 94.87 ms

3.3. Verification in other operating systems

In general, in other operating systems (Unix/similar/derivatives, Linux, BSD, etc.) the easiest thing to do is to use the ifconfig command, but in some cases graphical user interface environments exist that allow monitoring network interface and consequently IPv6 status (specific to each platform). Therefore, the examples specified for the case of Mac OS X are equivalent.

Likewise, ping6 and traceroute6 can also be used, for which reason the examples specified in the preceding section for Mac OS X also apply.

4. Advanced IPv6 Configuration

On certain occasions advanced configuration may be required, such as manually setting up or deleting IPv6 addresses.

Just as in the cases described above, these operations are performed differently in each operating system.

4.1. Advanced configuration in Windows

There are different reasons why it may be necessary to configure an IPv6 address manually. To do so, use the netsh command with the following format:
netsh interface ipv6 add address [interface=]<string (interface name or index)> [addrnetsh ess=]<IPv6 address>[/<integer>] [[type=]unicast|anycast] [[validlifetime=]<in teger>|infinite] [[preferredlifetime=]<integer>|infinite] [[store=]active|persistent]

Example:
netsh interface ipv6 add address 5 2001:db8: :2 type=unicast validlifetime=infinite preferredlifetime=10m store=active

Similarly, the configuration may be verified using netsh (assuming that the interface index is 5):
netsh interface ipv6 show address 5

Once an address is manually configured it can be modified as follows:
netsh interface ipv6 set address [interface=]<string> [address=]<IPv6 address> [[type=]unicast|anycast] [[validlifetime=]<integer>|infinite] [[preferredlifetime=]<intege r>|infinite] [[store=]active|persistent]

Example:
netsh interface ipv6 set address 5 2001:db8: :2 preferredlifetime=infinite

Finally, a manually configured address can be deleted as follows:
netsh interface ipv6 delete address [interface=]<string> [address=]<IPv6 address> [[store=]active|persistent]

Example:
netsh interface ipv6 delete address 5 2001:db8: :2 store=persistent

If we need to add a static route we can use the following:

netsh interface ipv6 add route add route [prefix=]<IPv6 address>/<integer> [interface=]<string> [[nexthop=]<IPv6 address>] [[siteprefixlength=]<integer>] [[metric=]<integer>] [[publish=]no|yes|immortal] [[validlifetime=]<integer>|infinite] [[preferredlifetime=]<integer>|infinite] [[store=]active|persistent]

Example:
netsh interface ipv6 add route 2001:db8: : 5 fe80: :200:87ff:fe28:a0e0 store=persistent
Where fe80: :200:87ff:fe28:a0e0 is the gateway we wish to configure for the 2001:db8: : route.

To delete the route:
netsh interface ipv6 delete route [prefix=]<IPv6 address>/<integer> [interface=]<string> [[nexthop=]<IPv6 address>] [[store=]active|persistent]

Example:
netsh interface ipv6 delete route 2002: :/16 5 fe80: :200:87ff:fe28:a0e0 store=persistent

Routes may be shown as follows:
netsh interface ipv6 show route [[level=]normal|verbose] [[store=]active|persistent]

Example:
netsh interface ipv6 show route

Publish	Type	Met	Prefix	Idx	Gateway/Interface Name
No	Manual	8	: :/0	13	Local Area Connection* 7
no	Manual	0	2002: :/16	5	fe80: :200:87ff:fe28:a0e0
no	Autoconf	8	2001:db8: :/64	5	Local Area Connection
no	Autoconf	256	: :/0	5	fe80: :200:87ff:fe28:a0e0

Finally, a DNS server can be added as follows:

netshinterfaceipv6adddnsserver[name=]<string>[address=]<IPv6address>[[index=]<entero>]

In Windows XP SP1/2003 SP1 dns is used instead of dnsserver.

Example:

netsh interface ipv6 add dnsserver "Local area network" 2001:7f9:1000:1: :947c 1

The index represents the position (preference) of the DNS server to be configured in the DNS server list.

Manually configured DNS servers can be shown using:

netsh interface ipv6 show dnsservers [[name=]string]

Example:

netsh interface ipv6 show dnsservers

DNS servers in LAN interface

Index	DNS server
1	2001:7f9:1000:1: :947c
2	2001:7f9:1000:1: :947c

And they can be deleted with:

netsh interface ipv6 delete dnsserver [name=]<string> [[address=]<IPv6 address>|all]

Example:

netsh interface ipv6 delete dnsserver "Local area network" all

4.2. Advanced configuration in Linux

Adding an IPv6 address:
/sbin/ip -6 addr add <ipv6address>/<prefixlength> dev <interface>
/sbin/ifconfig <interface> inet6 add <ipv6address>/<prefixlength>

Deleting an IPv6 address:
 # /sbin/ip -6 addr del <ipv6address>/<prefixlength> dev <interface>
 # /sbin/ifconfig <interface> inet6 del <ipv6address>/<prefixlength>

Adding a route through a gateway:
 # /sbin/ip -6 route add <ipv6network>/<prefixlength> via <ipv6address> [dev <device>]
#/sbin/route -A inet6 add <ipv6network>/<prefixlength> gw <ipv6address> [dev <device>]

Showing IPv6 routes:
 # /sbin/ip -6 route show [dev <device>]
 # /sbin/route -A inet6

Deleting a route through a gateway:
 # /sbin/ip -6 route del <ipv6network>/<prefixlength> via <ipv6address> [dev <device>]
 # /sbin/route -A inet6 del <network>/<prefixlength> [dev <device>]

Adding a route through an interface:
 # /sbin/ip -6 route add <ipv6network>/<prefixlength> dev <device> metric 1
 # /sbin/route -A inet6 add <network>/<prefixlength> dev <device>

Deleting a route through an interface:
 # /sbin/ip -6 route del <ipv6network>/<prefixlength> dev <device>
 # /sbin/route -A inet6 del <network>/<prefixlength> dev <device>

4.3. Advanced configuration in BSD

Adding an IPv6 address:
#>ifconfig <interface> inet6 add <dir. IPv6>

Deleting an IPv6 address:
#>ifconfig <interface> inet6 del <dir. IPv6>

For permanent configuration use the /etc/rc.conf file:
ipv6_enable="YES"
ipv6_ifconfig_rl0="2001:618:10:4: :4 prefixlen 64"

The /etc/defaults/rc.conf file may be used to check some of the existing options as well as those used by default.

To apply changes in rc.conf the computer must be restarted.

Adding a default route:
#>route –n add -inet6 default <dir. IPv6>

Deleting a default route:
#>route –n del -inet6 default

4.4. Advanced configuration in Mac OS X

Adding an IPv6 address:
 # ifconfig <interface> inet6 2001:db8:1:1: :2/64

Deleting an IPv6 address:
 # ifconfig <interface> inet6 delete 2001:db8:1:1: :2

Adding a default route:
 # route add -inet6 default [2001:db8:1:1: :1, -interface en1]

Deleting a default route:
#>route del -inet6 default

Showing IPv6 routes:
 # netstat -r -f inet6

5. IPv6 Transition Mechanisms

Because currenty not all ISP networks support IPv6, what we call transition and coexistence mechanisms must be used.

These mechanisms allow the coexistence of IPv4 and IPv6; even when IPv6 is not natively available, IPv6 can be used through the IPv4 network, mainly by means of what we known as tunneling.

Tunneling mechanisms allow IPv6 packets to be encapsualted in IPv4 packets, so that, as mentioned earlier, IPv6 can be transported over the existing IPv4 network.

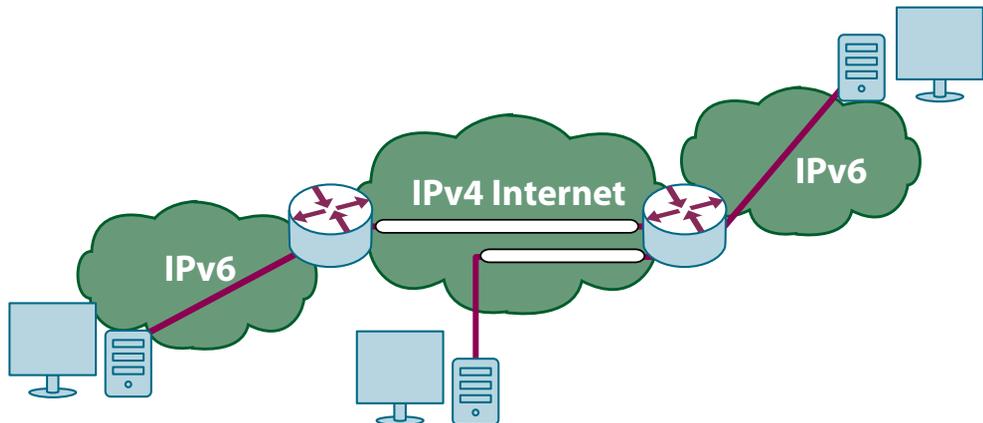The following images show how these tunnels work and how IPv6 is encapsulated in IPv4.

FIGURE 5: **IPv6 TUNNELING IN IPv4**

FIGURE 6: **IPv6 ENCAPSULATED IN IPv4**

Multiple transition mechanisms exist and this is quite a complex issue; for this reason, this section will focus only on the tunneling mechanisms we consider to be the most useful and which are known as automatic tunnels, more specifically, 6to4 and Teredo tunneling.

End Users

6to4 can only be used if public IPv4 addresses are available, for example, when a computer is connected to an ADSL network through a USB modem. Without going into technical detail, in this case what happens is that the IPv4 address is used to automatically configure an IPv6 address and an automatic tunnel which allows using IPv6 through the IPv4 network.

Instead, Teredo (or Miredo in Linux, BSD and Mac OS X systems) can use private IPv4 addresses, i.e. behind the so called "network address translators" or NAT, for example, when a connection to ADSL is established through a router instead of using a modem directly. Similar to the case of 6to4, an IPv6 address is automatically generated for each computer connected to that router/NAT and IPv6 is used through IPv4.

Because we are dealing with automatic transition mechanisms, generally no configuration is required and the operating system automatically detects whether the network has IPv6 connectivity (for example, provided by the ISP) and, if not, enables 6to4 or Teredo.

If Miredo is required, the user must simply download and install the software.

6. Uninstalling IPv6

It is not generally necessary to uninstall IPv6; however, the following information is provided in case for some reason is required to do so.

6.1. Uninstalling IPv6 in Windows XP/2003/Vista/7

In some of these platforms the following command may be used:
ipv6 uninstall

In other cases, as the ipv6.exe command was only available up to Windows XP, the netsh command must be used:
netsh interface ipv6 uninstall

Obviously it is also possible to use the graphical user interface, following the inverse steps as those indicated before for installing IPv6.

Usually the operating system must be restarted to avoid undesired effects.

Alternately, the following can be used to restore factory settings (in most platforms):
netsh interface ipv6 reset

Note that in Windows Vista, 2008 and 7, because the IPv6 stack is fully integrated with the IPv4 stack, it cannot be completely disabled. Instead, the graphical user interface can be used to disable it for each specific network interface.

6.2. Uninstalling IPv6 in Windows 2000

The procedure is as follows:
- Login to the system with local administrator privileges

- On the Windows 2000 desktop, click on Start, then Settings, then Network Connections. Alternately, right-click on My Network Places and then click on Properties.

- Right-click on Ethernet based connections for which you wish to add the IPv6 protocol and then click on Properties Usually this connection is called Local Area Connection.

- Select MSR IPv6 Protocol then click on Uninstall

- In the Uninstall MSR IPv6 Protocol dialogue box, clic on Yes

- In the Local Network dialogue box, clic on Yes to restart the computer

6.3. Uninstalling IPv6 in Mac OS X

We can disable IPv6 for all interfaces using: **#ip6 –x**

To re-enable IPv6: **#ip6 –a**

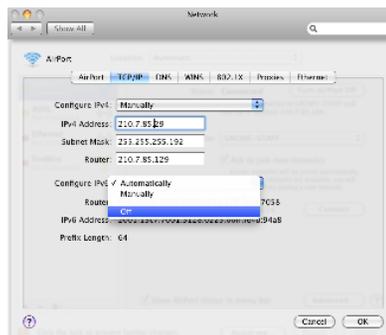Alternately, we can use the graphical user interface.

FIGURE 7: **DISABLING IPv6 IN MAC OS X**

3. Residential Networks and Home Offices

1. Introduction

1.1. What is a SOHO?

SOHO is the acronym for Small Office Home Office. In a more general sense, this term could be used to designate any office layout with a capacity of up to 10 workers[1] *(See Figure 1 and Figure 2).*

Based on this definition and as used in this document, IPv6 for Home Offices refers to the implementation of a SOHO networks such that it supports the new version of the IP protocol: IPv6.
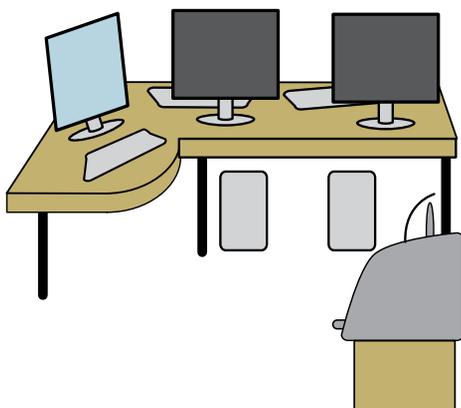
FIGURE 1: **EXAMPLE OF A SMALL BUSINESS HAVING LESS THAN 10 EMPLOYEES**
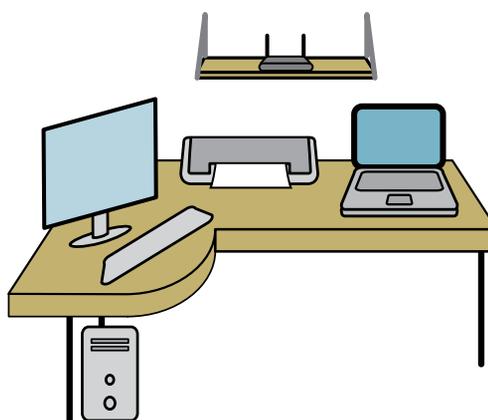
FIGURE 2: **EXAMPLE OF A HOME OFFICE OR RESIDENTIAL NETWORK**

1.2. Building a SOHO with IPv6 support

Before attempting to build a SOHO network with IPv6 support it is important to know the different parts that make up such a network. Once these components have been identified we will see which of them need to be configured so that they support IPv6. The final step is learning how to do it.

To summarize, we must complete the following steps:

1. Identify the components that make up the SOHO
2. Determine which of those components require configuration to support IPv6
3. Configure the SOHO for IPv6 support

---

1 http://es.wikipedia.org/wiki/Small_Office,_Home_Office

2. Identifying the Components that Make Up a SOHO

As mentioned earlier, this is the first step that needs to be considered when analyzing the construction of a SOHO network. We suggest performing this identification based on three distinct aspects:

2.1. Identifying the equipment that makes up the SOHO, also determining:
2.1.1. Networking devices
2.1.2. Terminal devices

2.2. Identifying operating systems:
1.2.1. Server operating systems
1.2. 2. Desktop and laptop computer operating systems

2.3. Identifying applications:
2.3.1. In servers
2.3.2. In terminals

Let's begin with item **2.1:**

- Networking devices: We must identify which devices are not part of our user interface but instead contribute to network communication. This group can include, among others, the switch to which terminals or computers are connected, the router that the Internet provider installs when we subscribed to the service, and the device that provides wireless connectivity.

- Terminal devices: This group includes all those devices with which we interact directly, such as desktop computers, portable or laptop computers, PDAs, IP telephones, application servers, among others.

We could add another category to include network printers, which are devices that will be connected to the network and may require IPv6 configuration even though they don't have a direct user interface and are not networking devices.

Continuing on to item **2.2**, we must identify the operating systems with which we will be using. We must consider:

- Server operating systems: Those that run on the terminal devices that provide network services such as email, including Linux, Windows, Unix, etc. Linux and Windows are the two operating systems most commonly used in SOHO networks.

- Desktop and laptop computer operating systems: Those that run on the terminals with which we work directly. In this case the most common operating systems are

Windows, Linux and MAC OS.

To complete the identification of the components that make up the SOHO network item **2.3.** requires that we identify:

- Server applications: Applications that provide services centrally for the different devices that make up the network, such as DNS, email, and web services, among others.

- Terminal applications: Applications that we use, for example, on our PDA, laptop or desktop computers. The most widespread terminal applications include text editors, spreadsheet applications, email clients, web browsers, instant messaging clients, multimedia services clients, custom applications, etc.

Now that we have clearly identified all the components of our SOHO network we can determine which of these components should be configured to support IPv6. We are ready to move on to the next step.

3. Determining which Components Require Configuration

In general, when dealing with a relatively new network (i.e. one with networking equipment manufactured no more than 3 to 4 years ago), all we need to do is update the operating systems that do not support IPv6.

A good practice would be to create a list of each networking device and search the available literature or documentation to determine their IPv6 compatibility. As already stated, if our equipment does not support IPv6 in its current condition, in order to provide IPv6 support some operating systems will have to be upgraded and/or firmware will need to be installed.

For example, Cisco routers support IPv6 from IOS version 12.3T and in the case of Juniper routers all JunOS versions support IPv6. Whether or not the wireless connection device will have to be configured to support the IPv6 protocol depends on the router's manufacturer and model. By way of an example, Apple AirPort[2] devices support IPv6, as do other devices such as D-link[3] wireless routers.

As regards operating systems, for several years most Linux and Unix versions have had the IPv6 stack enabled by default (for example, Solaris operating systems have provided IPv6 support since version 8). MacOS operating systems have supported IPv6 by default since 2003 with the "Panther" versions. Windows XP and Windows Server 2003 both offer a simple way to load the IPv6 stack. In Windows Vista this feature is enabled by default.

2 http://www.apple.com/airportextreme/specs.html,
3 http://www.ipv6ready.org/logo_db/logo_search2.php?logoid_number=01-000322&btm=Search

As to the applications, many will be independent of the IP protocol version that will be used, while others won't. In this case, the same criteria are applied as for the equipment: an evaluation of whether or not installed versions require an upgrade must be conducted. Custom-built applications will probably present the greatest problems and it may even be necessary to call the original developers to modify the code so that the applications can work independently from the IP protocol version that is used.

Within a context in which we will most likely encounter a network that combines devices that can be transitioned to IPv6 with others that cannot, we must bear in mind the recommendation to keep both versions of the protocol running simultaneously, in other words, use "dual-stack mechanisms".

4. Configuring IPv6 on SOHO Components

Finally, once all devices have been identified, all software versions have been upgraded to support the new version of the IP protocol, and all necessary modifications have been made to the applications, we are ready to move on to the configuration stage.

We will divide this task into two clearly separate areas:

• Configuring our SOHO's internal network (LAN)

• Configuring the external connection (Internet)

The following figure shows the boundary that separates both areas:

INTERNAL NETWORK   EXTERNAL NETWORK
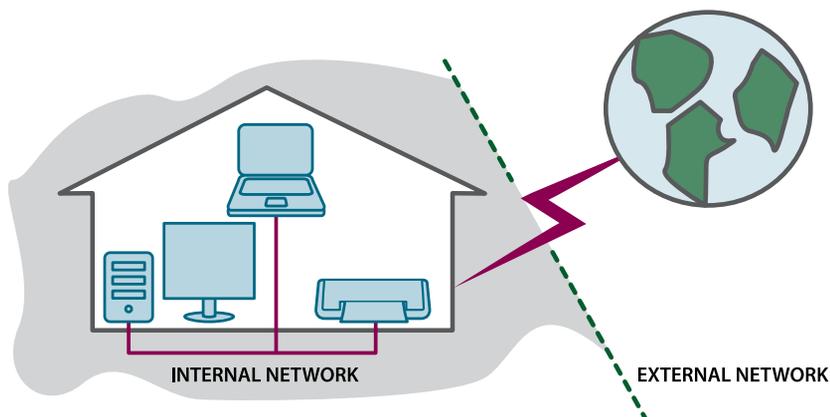
Before we begin describing both tasks we will discuss how to obtain the IPv6 addresses with which we will be working. Several alternatives are available, among them:

• Having our Internet provider assign us a prefix.

- Using tunnel brokers to establish automatic tunnels to a site capable of providing IPv6 connectivity. For this all we need is a dual-stack host and a browser that will allow us to see the broker's website or interface and from which to configure the tunnel.

Any of these options (or others not described in this book) will allow us to have IPv6 addresses, so we are now ready to start thinking about network configuration.

4.1. Configuring the internal network

Broadly speaking, there are two ways to configure our SOHO's internal network so that it will also work with IPv6: manual and automatic configuration (autoconfiguration).

Considering that the purpose of this book is to provide readers with a practical way to conduct their IPv6 experiment, we will focus on explaining how to configure our network automatically.

For IPv6 interface autoconfiguration it is necessary to request the corresponding configuration parameters and that another device announces those parameters.

Those requests and announcements are part of the Neighbor Discovery[4] Protocol which provides all the parameters needed for interface autoconfiguration through a set of ICMPv6[5] messages.

Without going into detail, the ICMPv6 messages that request the parameters are called "NS" (Neighbor Solicitation) and "RS" (Router Solicitation) messages; answers are contained in ICPMv6 messages called "NA" (Neighbor Advertisement) and "RA" (Router Advertisement) messages.

Now let's see how we can implement autoconfiguration in a SOHO network having a specific topology.

The example will be based on the network shown in *figure 3*.

As we can see, the SOHO network has a dedicated link, i.e. the Internet Service Provider assigns a connection for the exclusive use of that network. Typically, in those cases the Internet link connects to a router (R1 as shown in figure 3).

One of the router's interfaces is connected to the SOHO's internal network, where the remaining network devices are also connected. All connections are made through a switch, which in our example is called SW1.

---

4 http://www.ietf.org/rfc/rfc2461.txt
5 http://www.ietf.org/rfc/rfc2463.txt
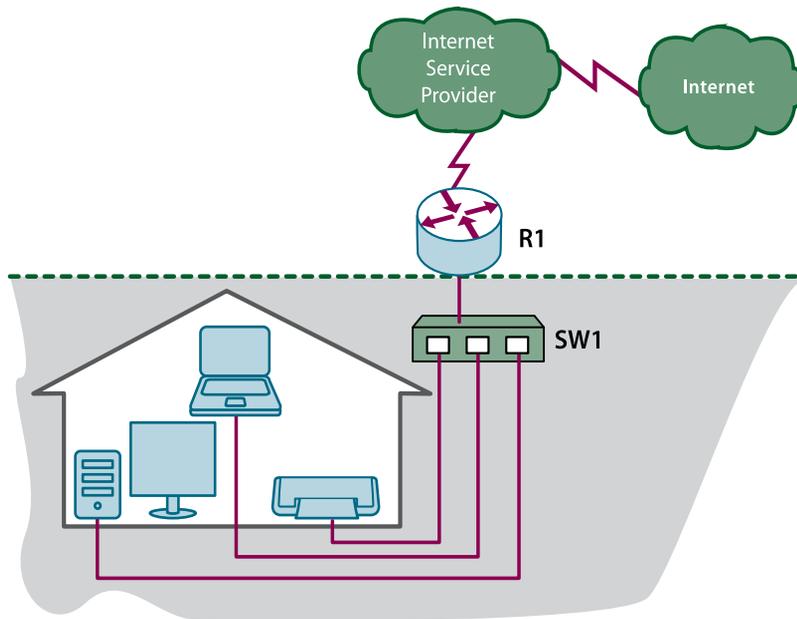
In this case, when a device (laptop computer, desktop computer, etc.) connects to the network, it sends an NS message so that all network nodes can see the device and generally it also sends an RS message. When the latter is received, router R1 replies with an RA message containing the IPv6 prefix that the device must use to complete the autoconfiguration mechanism. This message sequence is shown in the following diagrams:

New device connected to the network

IPv6 for All. A Guide for IPv6 Usage and Application.
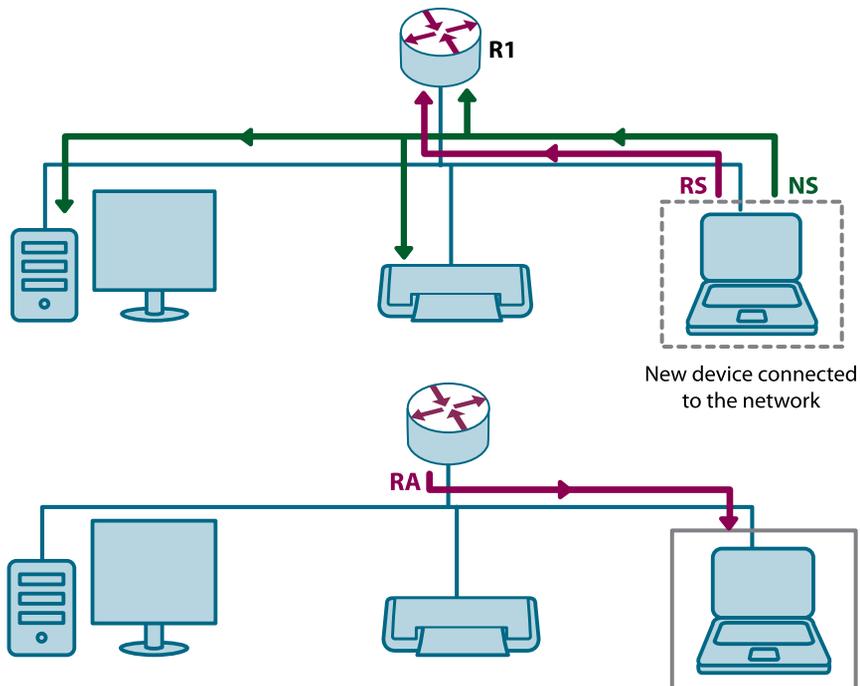
For example, in order for a Juniper router to know that it must announce the IPv6 prefix needed for the autoconfiguration of the internal network, the following command must be used[6]:

ipv6 nd prefix-advertisement <IPv6 prefix/IPv6 prefix-length>

In the case of Cisco routers, configuring an IPv6 address in the interface is enough for the router to start advertising the prefix to the internal network (prefix advertisement can be disabled if necessary).

Once the prefix has been obtained, the device can configure an IPv6 address based on the prefix advertised by the router and on its own MAC Address (using the EUI-64[7] method).

Figure 4 shows a model for obtaining IPv6 addresses in an an internal network after completing the autoconfiguration process.

2001: DB8 : FFFF : :

Mac Address
00:19:ab:b3:7c:db
IPv6 address obtained:
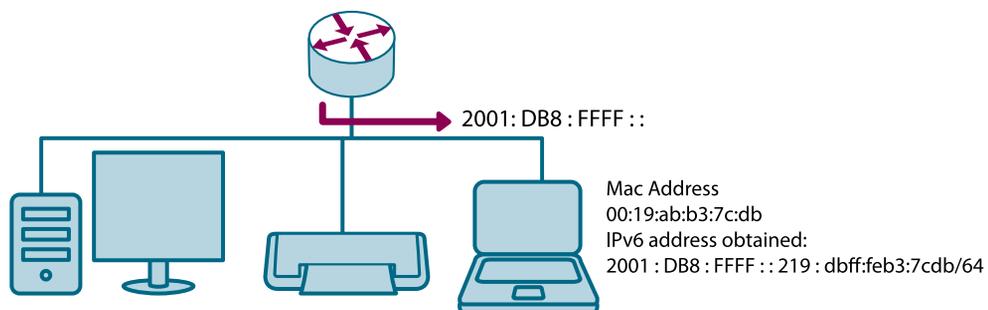2001 : DB8 : FFFF : : 219 : dbff:feb3:7cdb/64

FIGURE 4: **AUTOCONFIGURATION IN THE INTERNAL NETWORK**

Now let's suppose that we do not have access to the router that the Internet Service Provider installed when connecting our SOHO, or that this equipment simply does not exist. This means we must check who will be sending the RA messages.

One alternative is a computer connected to the internal network such that it announces the RA messages and therefore allows completing the autoconfiguration. For example, this could be a Linux server running the radvd[8] daemon. Another alternative is to use a DHCPv6[9] server *(see figure 5)*.

The difference between using the daemon or a DHCPv6 server lies in the fact that DHCPv6 can be used not only to announce network prefixes but also to communicate other data such as, for example, DNS server addresses, whereas the radvd daemon only

6 http://www.juniper.net/techpubs/software/erx/junose700/swcmdref-a-m/html/i-commands318.html
7 http://standards.ieee.org/regauth/oui/tutorials/EUI64.htm
8 http://en.wikipedia.org/wiki/Radvd
9 http://www.ietf.org/rfc/rfc3736.txt

announces IPv6 prefixes for interface autoconfiguration. However, in order to simplify network administration, it might be a good idea to use a combination of both methods.
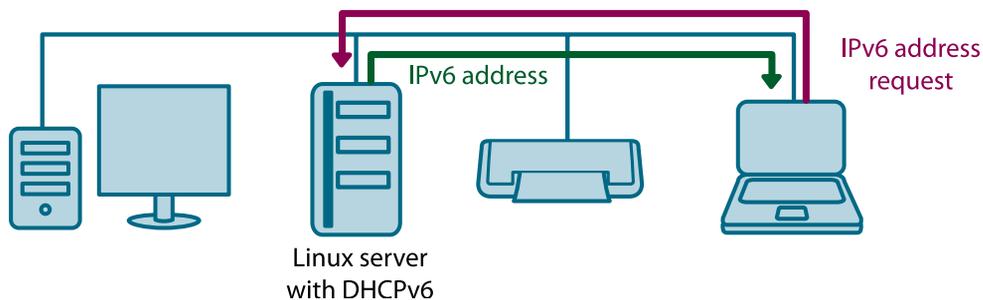
Figure 5: **Example of how to use a server to perform autoconfiguration**

In both cases, because we are talking about a SOHO, the IPv6 prefix is either assigned by the Internet Service Provider or corresponds to the SOHO's own addresses.

4.2. Configuring the external connection (Internet)

At this point it is highly likely that we have already decided how to configure the SOHO network so that it can internally operate using IPv6, in other words, the devices can already communicate through the LAN via IPv6.

This section describes the options available for configuring an IPv6 connection to the Internet.

As mentioned earlier, it is possible that the SOHO has a dedicated link and that the Internet Service Provider has installed a router that connects to the Internet.

Two possibilities exist:

A) In addition to the IPv4 connection, the ISP may provide a native IPv6 Internet connection.

B) The ISP cannot provide a native IPv6 Internet connection.

If our case is **A**, it is highly likely that the ISP is announcing its own IPv6 prefix on the Internet and that, if it provides this service to its clients, it will also offer us a prefix within its prefix. In this situation, if the ISP announces its prefix, it will also announce ours, which is a subset of its own address prefix.

The figure below shows this form of prefix assignment and announcement.

2001: DB8 : : /32

Internet

Internet Service Provider

2001 : DB8 : EEEE : : /48

Provider's router

2001 : DB8 : EEEE : FFFF : : 1/64

2001: DB8 : EEEE : FFFF : : 2/64
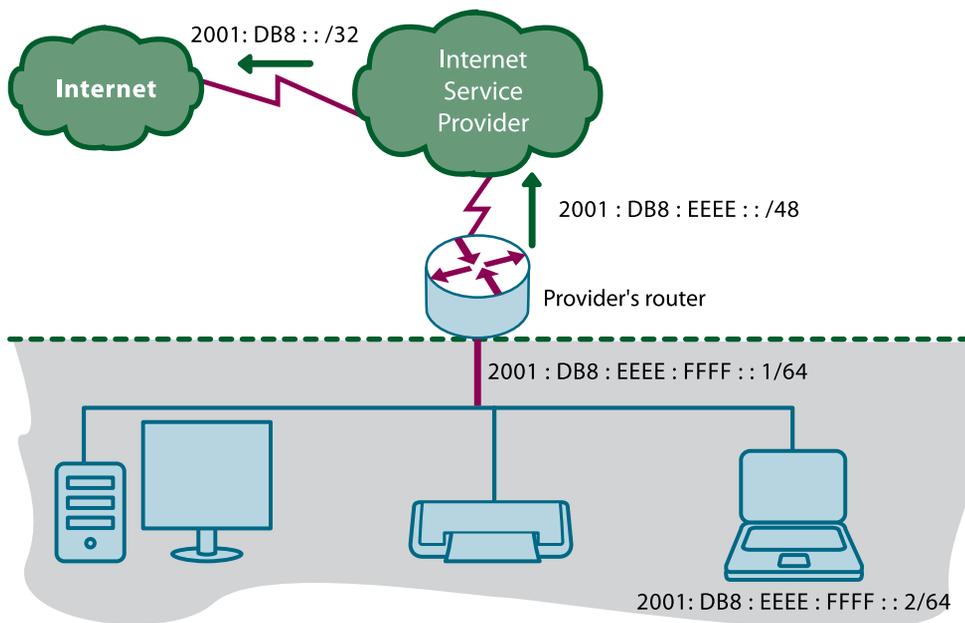
In this case all we need to do is talk to our Internet Service Provider to see what they prefer to do (use a BGP session with the SOHO network, use static routes to our router, etc.). Whatever the case may be, it is simply a matter of reaching an agreement. These agreements will depend on the different connectivity options available; however, it might be interesting to check RFC4779 and see which alternatives are best suited to our situation.

On the other hand, if our case is **B**, we will need to find a way to traverse the ISP's IPv4 network to reach another network that can understand and route our IPv6 packets. This requires some kind of tunneling mechanism.

Tunnels *(see figure 7)* are mechanisms that allow packets to be encapsulated so that they can traverse different networks. They can be classified into two major groups:

- Manual tunnels: As their name indicates, these tunnels are manually configured on both ends. Although it works, this solution requires establishing a static tunnel to a remote device that can provide connection to IPv6 networks.

- Automatic tunnels: As opposed to manual tunnels, automatic tunnels do not require static configuration at both ends; instead, they are established automatically and require minimum configuration.

4.2.1. Manual tunnels

This type of tunnel will not be described in great detail because, just as in the case of internal network configuration, we will focus on the alternative that is more practical from a user's perspective, i.e automatic tunnels.

As already mentioned, manual tunnels require configuration on both ends. The following figure shows how manual tunneling works.
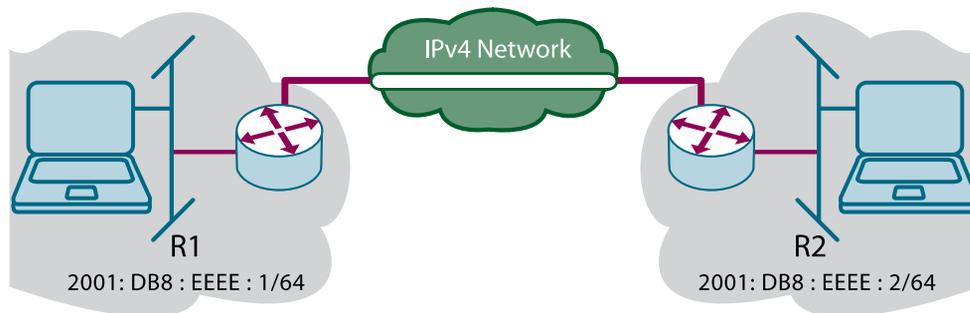
Figure 7: **Manual tunnel for traversing IPv4 networks**

A typical configuration for establishing the tunnel shown in the figure would be as follows:

In R1:
    interface SampleTunnelR1
    no ip address
    ipv6 address 2001:DB8:FFFF: :1/64
    tunnel source GigabitEthernet0/0
    tunnel destination 1.1.1.1
    tunnel mode ipv6ip

In R2:
    interface SampleTunnelR2
     no ip address
    ipv6 address 2001:DB8:FFFF: :2/64
    tunnel source GigabitEthernet0/1
    tunnel destination 2.2.2.2
    tunnel mode ipv6ip

These commands are general in nature and included only for illustrative purposes; their syntax may vary depending on the device that is to be configured, its manufacturer, operating system, type, etc. (the device may range from a router to a computer used as a router).

4.2.2. Automatic tunnels

Although we will describe the most relevant types of automatic tunnels, many other variants exist. Those we consider to be best suited to a SOHO network include 6to4 and Teredo tunnels.

4.2.2.1. 6to4 tunnels

6to4 tunneling is a mechanism that allows IPv6 devices connected only to IPv4 networks to reach other IPv6 networks. This is done using a set of addresses assigned by the IANA[10] for 6to4 tunneling: prefix 2002: :/16.

The 6to4 tunneling mechanism works as follows. A device that has an IPv6 address wishes to communicate with another IPv6 address outside its own network. To do so it must have access to a router that supports 6to4 pseudo-interfaces and that is able to route the 2002: :/16 prefix.

In addition, it requires at least one public IPv4 address based on which to calculate the 6to4 address for the router. This calculation is performed as follows:

1- The IPv4 address is expressed in nibble notation, for example:

In nibble notation, IPv4 address 192.0.2.1 is expressed as follows:
   192 ----> C0
   0 ---> 00
   2 ---> 02
   1 ---> 01

2- The first part of the router's address is built using the 6to4 address prefix mentioned above as follows:
   2002:C000 : 0201: :/48

3- Now that we have the prefix for our router we can choose any interface identifier[11], for example:
   2002:C000:0201: :1/128

Continuing with the description of how 6to4 tunneling works, in addition to the device attempting to communicate with an IPv6 network and the router (typically a border router) with the 6to4 pseudo-interface, we need an Internet router with which to establish the 6to4 tunnel. But how do we find that router? There are several of these devices available on the Internet, all of which have the anycast address 192.88.99.1[12]. Using nibble notation, this address would be 2002:c058:6301: :/128.

A tunnel will be built between our router's IPv4 addresses and anycast address 192.88.99.1. This means that we will have a 6to4 IPv6 prefix – 2002:C000:0201: :/48 – to

---

10 http://www.iana.org/
11 http://www.ietf.org/rfc/rfc3513.txt
12 http://www.ietf.org/rfc/rfc3068.txt

use on our LAN and 2002:c058:6301: :/128 will be reachable through the tunnel. Prefix 2002: :/16 must be routed through this interface.

The following is an example of how to create a 6to4 tunnel on a Cisco router:

interface Tunnel2002
description Tunnel 6to4 to Internet
no ip address
no ip redirects
ipv6 address 2002:C000:0201: :/48
tunnel source GigabitEthernet0/0
tunnel mode ipv6ip 6to4

interface GigabitEthernet0/0
description 6to4 interface
ip address 192.0.2.1 255.255.255.0

ipv6 route 2002: :/16 Tunnel2002
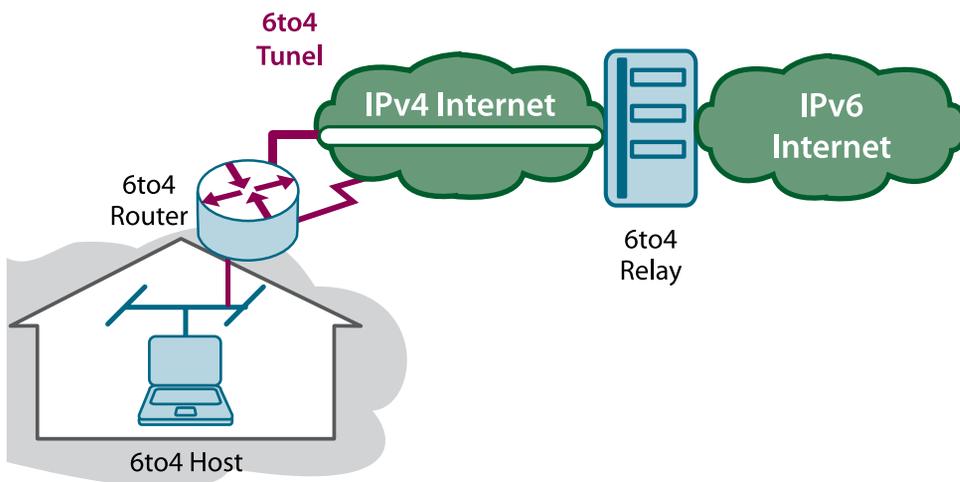
FIGURE 8: **ESTABLISHING A 6TO4 TUNNEL**

4.2.2.2. Teredo[13] tunnels

Teredo (or Miredo in the case of open-source software) is a mechanism that allows a device to access IPv6 networks even when behind an IPv4 NAT.

To do so, a server (e.g. a Linux or BSD server) must be available to provide the IPv6 addresses with which we will be able to traverse NAT. The server must have a public IPv4 address and be reachable through the Internet.

13 http://www.ietf.org/rfc/rfc4380.txt

A Teredo Client is a device that attempts to access the Internet through this Teredo Server and connect to an IPv6 address.

A Teredo Server listens for Teredo Client requests on UDP port 3544[14] and returns an IPv6 address for the Client to use to reach its destination.

In order for the traffic to be able to flow to and from Internet IPv6 addresses and our own Teredo Client we will communicate using a Teredo Relay. This Relay is in charge of receiving and forwarding the Teredo Client's IPv6 traffic.

The Teredo Server will also announce Teredo prefix 2001:0000: :/32 on the Internet.
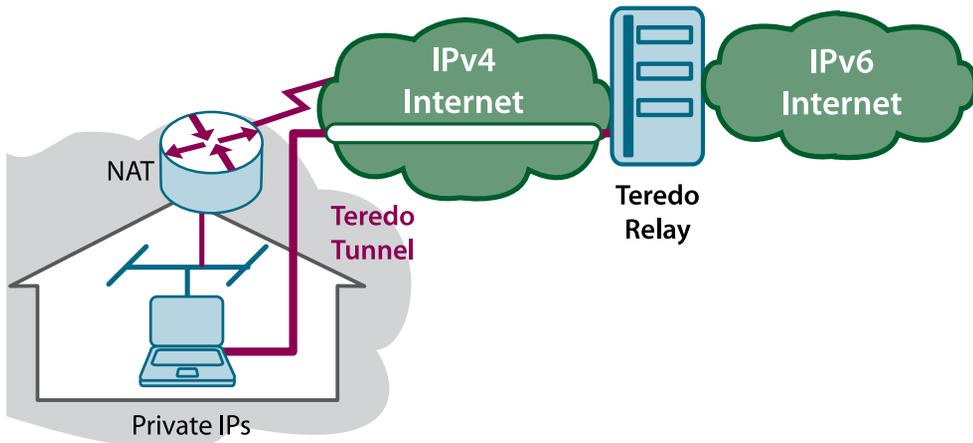
FIGURE 9: TEREDO/MIREDO TUNNELING

If you have reached this point and completed all the steps described above, you now have IPv6 connectivity both internally as well as with the Internet. If so, we can say that our mission is accomplished.

> We would like to take this opportunity to remind the reader that the object of this chapter is to provide a guide to the tools and steps that may help build a small office or residential network so that it supports IPv6, but that the tools and steps described herein are not the only options available. On the contrary, what we have shown here is but a small subset of the practical aspects relating to IPv6 transition mechanisms.

5. References

http://portalipv6.lacnic.net/
http://www.ipv6tf.org

---

14 http://www.ietf.org/rfc/rfc0768.txt

4. IPv6 Services

1. Introduction

This chapter describes how to install and configure several basic IPv6 services on some of the most common operating systems. Practically all widespread services, applications and devices support IPv6 (for a detailed list, please see *http://www.ipv6-to-standard.org*).

It must be understood that the service application or software runs on a platform –the server– that has its own operating system and hardware. Thus, the first step is to enable IPv6 on the server's operating system to allow data to be transported to/from the server using the IPv6 protocol. The steps for enabling IPv6 on different operating systems can be found in the End Users chapter. After that, all we must do to deploy IPv6 services is install and configure the packages that support IPv6 and which are generally extended versions of those that support IPv4.

2. About the Services

The services offered over the Internet are designed to be accessed by any client, in other words, the client-server model is based on a server that is accessed by many clients. In this model communications are always initiated by the client.

To access the services, the client must know the server's network or IP (Internet Protocol) address. The Domain Name System (DNS) was created with the aim of making this process more end-user friendly. This system translates a server's domain name into the server's IP address and therefore the name can be used instead of the corresponding IP address.

For example, when a user accesses www.google.com from a web client, this domain name is resolved to the IP address of a web server that seamlessly offers the Google website to the client.

The IP address obtained through the DNS may be an IPv4 address, an IPv6 address, or both. This allows accessing services via IPv6 in a user-friendly and transparent manner.

The following sections describe how to install and configure different services on different platforms.

3. Telnet

3.1. Description of the service

Telnet is a well-known application used to communicate with another device using a command interface with the TELNET Protocol through TCP port 23. It is based on the client-server model and therefore both are required to establish the communication. The telnet server is installed with the telnetd package.

3.2. Installation and configuration steps

Different versions of the package exist for different Linux distributions, the most common of which are installed as follows.

3.2.1. Debian:

To install the service use:
 # sudo apt-get install telnetd

The configuration file is /etc/inetd.conf.

To restart the service use:
 # sudo /etc/init.d/inetd restart

3.2.2. Fedora:

To install the service use:
 # yum install telnet-server telnet

Telnet is installed as a service invoked by the xinetd process. To enable or disable telnet the /etc/xinetd.d/telnet file must be modified. To enable telnet: disable = no.

To restart telnet use:
 # /etc/init.d/xinetd restart

3.2.3. Red Hat Enterprise:

To install the service use:
 # up2date telnet-server telnet

Telnet is installed as a service invoked by the xinetd process. To enable or disable telnet the /etc/xinetd.d/telnet file must be modified. To enable telnet: disable = no.

To restart telnet use:
 # /etc/init.d/xinetd restart

3.2.4. Ubuntu:

To install the service use:
 # sudo apt-get install telnetd

The configuration file is /etc/inetd.conf.

To restart the service use:
 # sudo /etc/init.d/openbsd-inetd restart

3.2.5. FreeBSD:

In FreeBSD the telnet server package is already installed by default in /usr/libexec/telnetd.

The configuration file is /etc/inetd.conf. To enable the telnet server, remove the comment (i.e. delete the # character) from the following line:
#telnet stream tcp nowait root /usr/libexec/telnetd telnetd

Then the inetd daemon must be enabled. In file /etc/rc.conf, add the following line:
inetd_enable= "YES"

To conclude, restart the telnet server through the inetd service using the following command:
 # /etc/rc.d/inetd restart

4. SSH

4.1. Description of the service

SSH allows communicating with another device using a comnand interface with a secure encrypted channel through TCP port 22. Usually SSH is used instead of telnet when secure communication is required. SSH is also based on the client-server model and therefore both are required to establish the communication. The SSH server is installed with the sshd package.

4.2. Installation and configuration steps

Several SSH server applications exist. The most common SSH application for Linux is Portable OpenSSH, while the most common SSH application for BSD is OpenSSH.

4.2.1. Debian/Ubuntu:

To install the service use:
 # sudo apt-get install openssh-server

After its installation the SSH server is enabled by default. To stop, start or restart the SSH server use:
 # sudo /etc/init.d/ssh stop
 # sudo /etc/init.d/ssh start
 # sudo /etc/init.d/ssh restart

4.2.2. Red Hat Enterprise:

The openssh-server-4.3p2-29.el5.i386.rpm package or later includes an SSH server (http://rpmfind.net). To install it use:
 # rpm -ihv openssh-server-4.3p2-29.el5.i386.rpm

The server has two configuration files: /etc/ssh/sshd_config and /etc/ssh/ssh_host_key. The first file is used to configure general aspects and, while it may be modified to adapt it to a particular system, its default installation parameters are usually enough to use the SSH server. The second file is used to store the keys used when communicating with other hosts.

Alternately, the following command can be used to search for an sshd and install it if necessary:
 # up2date --showall | grep sshd

4.2.3. FreeBSD:

OpenSSH is part of the operating system core, therefore no installation steps are required. The service is enabled in /etc/rc.conf

5. FTP

5.1. Description of the service

The FTP protocol is used to transfer or obtain files to/from a remote host. FTP generally uses ports 20 and 21. It is based on the client-server model and therefore both are required to establish the communication. The FTP server is installed with the ftpd package.

5.2. Installation and configuration steps

Several FTP server packages support IPv6 (http://linuxmafia.com/faq/Network_Other/ftp-daemons.html). Some of the most common packages are installed as described below.

5.2.1 Red Hat:

The Pure-FTPd package can be installed from pure-ftpd-1.0.22.tar.gz or later (http://www.pureftpd.org). To install the service use:
 # tar xzvf pure-ftpd-1.0.22.tar.gz

Go to the resulting folder and execute the typical installation commands:
./configure
make
make install

5.2.2. Ubuntu:

To install the proftpd package use:
 # sudo apt-get install proftpd

6. Email

6.1. Description of the service

Email is one of the most widely used services. Generally, emails are sent using SMTP protocols (port 25) and fetched using POP3 (port 110) or IMAP4 (port 143). The service is based on the client-server model and therefore both are required to establish the communication. Most common email servers and clients support IPv6.

6.2. Installation and configuration steps

Several SMTP, POP3 and IMAP4 packages support IPv6. Sendmail (http://www.sendmail.org) is a very popular SMTP server for Unix environments. Washington University or WU-IMAP (http://www.washington.edu/imap) is widely used for IMAP4 and POP3 servers. The following sections describe how to install and configure these packages in various operating systems.

6.2.1. Linux:

Download and install Sendmail.

In Sendmail IPv6 is not enabled by default (at least not in versions earlier than 8.12.X). To enable IPv6 support, go to configuration file devtools/Site/site.config.m4 and add the following line:
APPENDDEF(`confENVDEF', `-DNETINET6')

Rebuild Sendmail.

Then, go to file sendmail.mc and add the following line:
DAEMON_OPTIONS(`Port=smtp, Name=MTA-v6, Family=inet6')dnl

Create a new sendmail.cf and restart Sendmail.

If an error message is displayed make sure that the related libraries support IPv6 by rebuilding them with IPv6 support.

If you want to use a POP3/IMAP4 server, download and install UW-IMAP.
To enable IPv6 support, go configuration file /etc/inetd.conf and add the following lines:

```
# IMAP server with IPv6 support
imap stream tcp6 nowait root /usr/sbin/tcpd imapd
# POP3 server with IPv6 support
pop-3 stream tcp6 nowait root /usr/sbin/tcpd ipop3d
```

Courier-IMAP can be used instead of UW-IMAP. Download and install Courier-IMAP (http://www.courier-mta.org/imap).

The Courier-IMAP building process will automatically detect whether or not the operating system supports IPv6 and, if so, enable IPv6 support. This means that no additional steps should be required.

6.2.2. FreeBSD:

Download and install Sendmail.

To enable IPv6 support, go to configuration file etc/sendmail.ipv6.cf and add the following line:

```
# SMTP daemon options
O DaemonPortOptions= Port=smtp, Name=MTA-v6, Family=inet6, Addr=[email
    server IPv6 address]
```

To start the Sendmail service, go to file /etc/rc.local and add the following line

```
# SMTP Sendmail server with IPv6 support
/usr/sbin/sendmail -C/etc/sendmail.ipv6.cf -bd -q30m
```

Popper is another POP3 server. To install it in BSD use:

```
# cd /usr/ports/mail/popper
# make install
```

To enable Popper with IPv6 support, go to file /etc/inetd.conf and add the following line:

```
# POP3 Popper server with IPv6 support
pop3  stream  tcp6  nowait  root  /usr/local/libexec/popper  popper
```

6.2.3. Windows Server 2008:

Windows Server 2008 has full IPv6 support for all main network applications and services except Internet Information Services (IIS) SMTP servers. However, Microsoft Exchange Server 2007 with Service Pack 1 supports IPv6 for SMTP. In general terms, the installation and configuration of the Exchange Server for IPv6 is the same as for IPv4.

7. Multimedia Streaming

7.1 Description of the service

The need to transmit or stream audio and video through the Internet and/or Intranets is becoming increasingly common. Multimedia streaming is based on the client-server model and therefore both are required to establish the communication.

7.2. Installation and configuration steps

Several multimedia content streaming programs exist that suppot IPv6. Windows Media Services is the most common streaming media server for Windows platforms. It is installed and configured as follows.

7.2.1. Windows Servers

In the case of Windows Servers 2000, 2003 and 2008, Windows Media Services (WMS) can be used to stream live or on-demand audio and video. Windows Media Services acts as a streaming media server from encoded sources. The tasks performed by Windows Media Services include waiting for client requests; checking if a specific user is allowed to connect; controlling network connections; building streaming packets using the encoded sources as payload; delivering the streaming packets with IPv4 and IPv6 to unicast, anycast and multicast destinations; etc.

- Windows Media Services is a component integrated into Windows Server operating systems.

- If necessary, in Windows Server 2003 it can be updated using:
  http://download.microsoft.com/download/1/2/e/12e25064-8b99-4229-a554-acb67493742d/UpgradeWMS9S.exe

- Windows Server 2008 also includes Windows Media Services 2008; if not, it can be installed using:
  http://www.microsoft.com/windows/windowsmedia/forpros/serve/prodinfo2008.aspx

Another application that may be required for encoding multimedia sources is Windows Media Encoder (WME). This application encodes multimedia sources such as DVD, analog audio/video inputs, etc. into formats that can be used for streaming, such as mp3 in the case if audio or AVI in the case of video. Windows Media Encoder can also be used for multimedia streaming, but only for a limited number of clients (5 or less). To install this application use *http://download.microsoft.com/download/8/1/f/81f9402f-efdd-439d-b2a4-089563199d47/WMEncoder.exe*

To access the Windows Media Services configuration interface go to Programs > Administrative Tools > Windows Media Services.

Multimedia content streaming is based on publishing points.

7.2.1.1. Creating a new publishing point

Two different models exist: **Push** and **Pull**.

Push

The encoder starts multimedia streaming. The location of the streaming server is configured on the encoder so that each time encoding begins the multimedia flow is sent to that particular streaming server. This is the easiest way to handle this, although the encoder and streaming server use up a lot of bandwidth even when there are no users connected to the server.

Configuration is as follows:

- In the encoder, go to Properties > Output, select "Push to server (the connection is initiated by the encoder)"
  - Server name: streaming.example.com:8100 (this is the streaming server at port 8100, as port 80 may be busy with with a web server)
  - Publishing point: name_of_the_event_to_be_streamed (this will be the publishing point for the users' connections)
  - Copy settings: push_test (this is a setting of the streaming server that is copied to create the "name_of_the_event_to_be_streamed" publishing point. This setting extracts the streaming from the encoder and enters push:* under streaming type).

- Compression may be adjusted under Properties > Compression. For testing purposes it is recommended that the total streaming bandwidth does not exceed 150 kbps. Larger bandwidths may be used depending on resource availability on the networks where the streaming is to take place.

- After this has been completed, "Start Encoding" is activated and the encoder sends the flow to the streaming server. A notification may appear regarding the steps that must be followed if the publishing point is multicast. This can be ignored.

- At one point the encoder will require a username and password to be published on our streaming server. When prompted, enter the following username/password: test/4321.

- The user must have WMS writing privileges. This is configured under Properties (of the streaming server) > Authorization > WMS publishing points ACL authorization.

- We must also enable Properties (of the streaming server) > Authentication > WMS negotiation authentication.

- Thus, the server will automatically create the publishing point and users will be able to connect to this point through the following URLs:
  - *http://streaming.example.com:8100/name_of_the_event_to_be_streamed*
  - *mms://streaming.example.com/name_of_the_event_to_be_streamed*

- The publishing point is for distribution (not for on-demand streaming) and is displayed in the graphic interface in blue, not green.

Pull

Configuration is as follows:
- In the encoder, under Properties > Output, select "Pull"

- A pull-type publishing point is configured on the WM Server with an URL such as
  - *http://server_name:server_port*

- When the WM Server receives a connection request from a user, the server connects to the encoder and begins streaming.

7.2.1.1. Event streaming/recording

For performance reasons, a dedicated server is usually required (Windows 2003) for installing Windows Media Encoder (WME).

If an external video camera is used, a video capture card connected to the camera must also be installed on that server. Alternately, a USB camera may be used without a video capture card.

WME can be configured to encode audio/video captured by the camera, stream the content and also record it to the local hard drive. If more than five streaming connections are expected, then more than one device will be required: one computer for WMS and another one for WME. This depends on the CPU capacity of the server recording the session. To use a streaming server plus an encoder follow the instructions provided for the Pull model above.

8. Web

8.1. Description of the service

Web browsing uses the HTTP protocol to transfer hypertexts, web pages or HTML pages. It generally uses port 80 and is based on the client-server model, therefore both are required to establish the communication. The web or HTTP server is installed with the httpd package.

8.2. Installation and configuration steps

Many programs are used to provide this extremely popular service, but the most common are Apache and IIS. We will now describe how to install and configure both applications so that they will reply to requests over IPv6.

8.2.1. Apache

Apache is the most widespread web server in use today and it is usually run on Linux platforms. IPv6 support is available starting from versions 2.x. The following examples are based on version 2.0.63.

The installation can use each distribution's usual package management utility (apt-get install apache2, yum, up2date, rpm, etc.); alternately, the source files can be downloaded from http://httpd.apache.org and compiled:

```
#>cd /usr/local/src
#>tar -xzvf httpd-2.0.63.tar.gz
#>cd httpd-2.0.63
#>./configure --prefix=/usr/local/apache2 --enable-module=so
#>make
#>make install
```

The --prefix parameter specifies the folder where the server will be installed. The --enable-module=so parameter enables Dynamic Shared Object (DSO) support to allow dynamically loading modules, for example PHP.

8.2.2.1. Listening on IPv6

IPv6 support is enabled by default starting from Apache 2.0.x. Therefore, after it is installed and started, it will listen for IPv6 connections. Remember that IPv6 must be previously enabled on the Linux server.

The directive that controls the IPs and ports on which the web server listens is Listen. This directive is found in the main configuration file, httpd.conf. By default it listens on all IPs and port 80 (http):

```
Listen 80
```

The netstat command can be used to verify that the server at port 80 is listening on IPv6:

```
[root]# netstat -tan
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address  Foreign Address State
...
tcp  0   0   : ::80    : ::*       LISTEN
...
```

This indicates that it is listening (LISTEN) on any server address (: :), be it IPv4 or IPv6, on port 80 (:80).

8.2.1.2. Virtual hosts

In order to configure virtual hosts, square brackets [ ] must be used around the IPv6 address, as in the following example:

NameVirtualHost [2001:db8:1: :1000:1234]
NameVirtualHost 10.0.0.3

<VirtualHost [2001:db8:1: :1000:1234]>
 DocumentRoot /example/htdocs/web-v4-v6
 ServerName www.example.com
</VirtualHost>

<VirtualHost 10.0.0.3>
 DocumentRoot /example/htdocs/web-v4-v6
 ServerName www.example.com
</VirtualHost>

<VirtualHost [2001:db8:1: :1000:1234]>
 DocumentRoot /example/htdocs/web-v6-only
 ServerName ipv6.example.com
</VirtualHost>

The configuration above allows the server:

- To answer IPv4 requests on address 10.0.0.3 and IPv6 requests on address 2001:db8:1: :1000:1234

- Requests received on those addresses can be be distinguished by the URL to which they refer, therefore

- Requests for *www.example.com* will be answered via IPv4 and IPv6, serving the contents of the /example/htdocs/web-v4-v6 folder

- Requests for ipv6.example.com will be answered via IPv6 only, serving the contents of the /example/htdocs/web-v6-only folder

NOTE: In the example above, www.example.com would typically resolve to both addresses, IPv4 and IPv6. Likewise, ipv6.example.com will resolve only to the IPv6 address. For further information please see the DNS section of this chapter.

6.1.3. Tip: Displaying the client's IPv6/IPv4 address

It might be interesting to display the source IP address used by the client to access our webpage. Although there are several ways to do this, we will show a way to do it using PHP, the most common programming language in Linux/Apache environments.

Simply include the following code in the initial page, for example index.php:

```
<?php if(strpos($_SERVER['REMOTE_ADDR'],".")===false)
 {
  echo "<font color='#FF0000' size=2 face='verdana'>You are using IPv6 (".$_
SERVER['REMOTE_ADDR'].").</font><br><br>";
  }else{
  $DIRv4=str_replace(": :ffff:","",$REMOTE_ADDR);
   echo "<font color='#FF0000' size=2 face='verdana'>You are using IPv4 (".$_
SERVER['REMOTE_ADDR'].").</font><br><br>";
  }
?>
```

8.2.1.3. Tip: Disabling sendfile

Apache 2 supports a method called sendfile offered by the operating system that increases the speed with which data is delivered. Some network card controllers also support performing offline TCP checksums. In some cases this can lead to connection issues and TCP checksums that are invalid for IPv6 traffic.

In these cases sendfile must be disabled or, alternately, the server must be compiled again using the **--without-sendfile** option or using the **EnableSendfile off** directive in the Apache configuration file (httpd.conf).

The EnableSendfile off directive is only supported in versions later than 2.0.44.

8.2.1.5. Verifying that it works

Using a browser on the same server we can check the address through which we are accessing the server to verify that access is possible through IPv4 and IPv6. To do so localhost IPv4 (127.0.0.1) and IPv6 (: :1) addresses may be used.
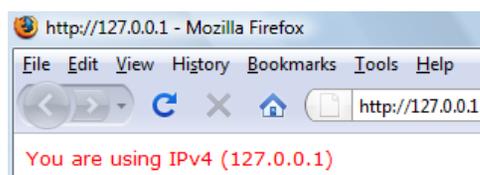
8.2.2. IIS

Microsoft IIS (Internet Information Services) runs on Windows servers, which is why here we will explain its configuration based on Windows Server 2003 R2 SP2 Standard Edition which includes IIS v6.0.

It can be installed/uninstalled using **Add or Remove Programs** in the control Panel. **Add/Remove Windows Components** allows accessing the Windows Components Wizard.
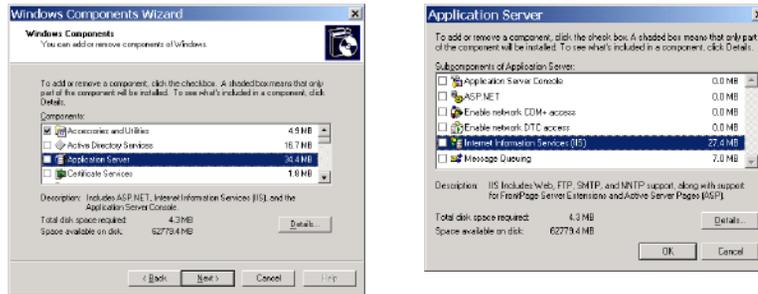
In the Windows Components Wizard, select **Application Server** and click on Details… Select **Install Internet Information Services (IIS)** and click on **Details…**
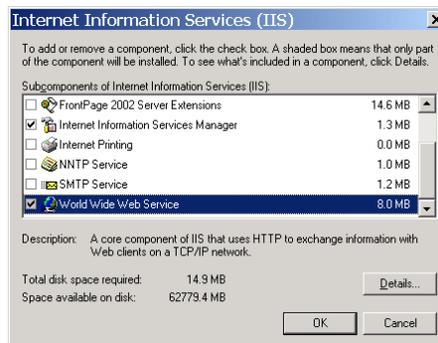
Proper installation of the web server requires enabling **Internet Information Services Manager, Common Files and World Wide Web Service.**

NOTE: A Windows Server 2003 installation CD will be required.

8.2.2.1. Listening on IPv6

Once the IIS server and IPv6 *(C:\>netsh interface ipv6 install)* are installed, restarting the IIS service is recommended to ensure that it listens on IPv6. This is done using the **IIS Manager** found under Administrative Tools. Right-clicking on the server that hosts the IIS displays the option **Restart IIS…** under All Tasks:
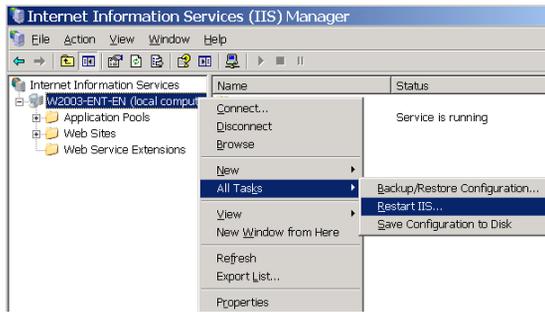
We can check that it is listening on port 80 (http) on IPv6:
C:\>netstat -an -p tcpv6
Active connections

Proto	Local address	Remote address	Status
TCP	**[: :]:80**	**[: :]:0 LISTENING**	**0**
...			

8.2.2.2. Configuring IIS

IPv6 must be enabled individually for each website hosted on a particular IIS installation. This is done using the **IIS Manager** found under Administrative Tools. Website properties are configured by right-clicking on each website and selecting Properties:
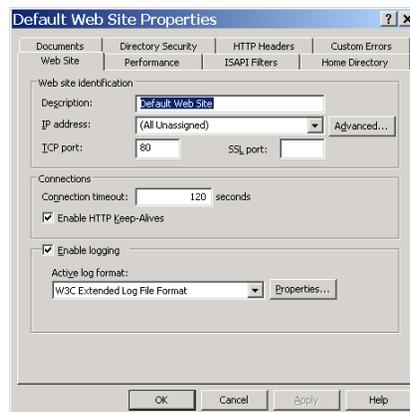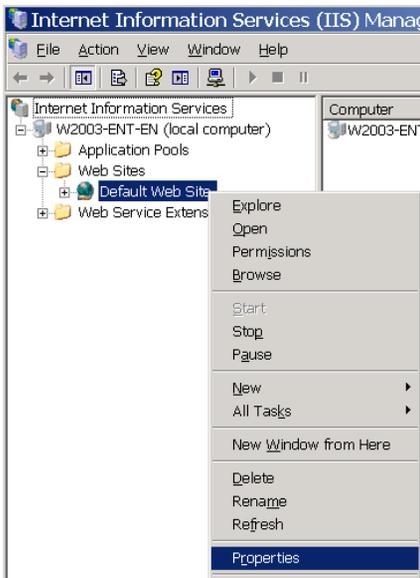
On the Website tab, under IP Address, the option None Assigned must be selected. This allows listening on port 80 and on all IPv4 and IPv6 addresses. Details can be added by clicking on Advanced...

The following figure shows an example in which the website can be accessed through:
- **IPv4 only:** *ipv4.example.com*, which resolves to IPv4 address 192.168.1.101
- **IPv4 and IPv6:** *www.example.com*, which resolves to the server's IPv4 and IPv6 addresses
- **IPv6 only:** ipv6.example.com, which resolves to the server's IPv6 address
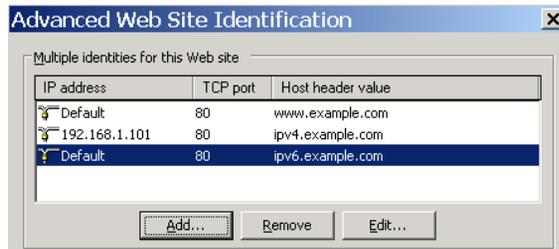
The following is another, less complex example that would allow accessing the website using any IP and any domain name:
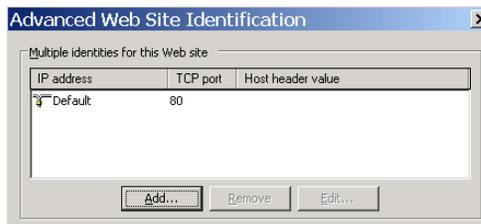
8.2.2.3. Tip: Displaying the client's IPv6/IPv4 address

It might be interesting to display the source IP address used by the client to access our webpage. The following example shows how to do this using ASP, the most common programming language in Windows/IIS environments. Simply include the following code in the initial page, for example default.asp:

```
<%
    if InStr(Request.ServerVariables("REMOTE_ADDR"),".") = 0 then
     response.Write( "<font color='#154983' size=2 face='verdana'> You are using
IPv6.<br><br>")
     else
     response.Write ("<font color='#FF0000' size=2 face='verdana'> You are using
IPv4.<br><br>")
     end if


     response.Write ("(" &Request.ServerVariables("REMOTE_ADDR") & ")</
font><br><br>")
     %>
```

NOTE: For ASP pages to work, they must be enabled under Web Service Extensions on the IIS Manager, as shown in the following figure.
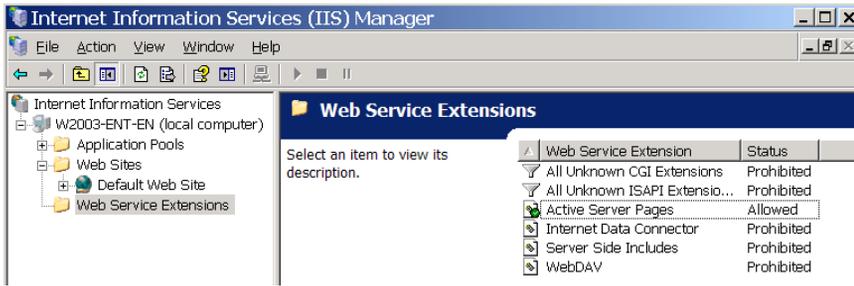
8.2.2.4. Verifying that it works

Using a browser on the same server we can check the address through which we are accessing the server to verify that access is possible through IPv4 and IPv6. To do so localhost IPv4 (127.0.0.1) and IPv6 (::1) addresses may be used.
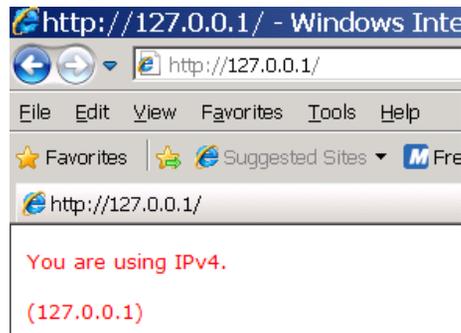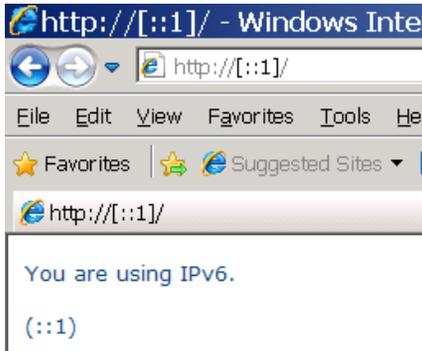
9. DNS

9.1. Description of the service

The DNS service translates domain names into both IPv4 and IPv6 network addresses. It plays a key role in the Internet as we know it today.

Without going into detail about how DNS works, it should be clear that transporting DNS traffic (through an IPv4 and/or IPv6 network) and the data contained in DNS responses (A records for IPv4 and AAAA records for IPv6) are two separate things. Both are independent of the IP protocol that is used. Figure 1 shows how IPv4 and IPv6 transport can be used indistinctly to resolve an IPv6 address (AAAA).

For the reason stated above, we will see both how the server application is configured to answer IPv6 requests (transport) as well as how this IPv6-related data is included in the content that is served (data).
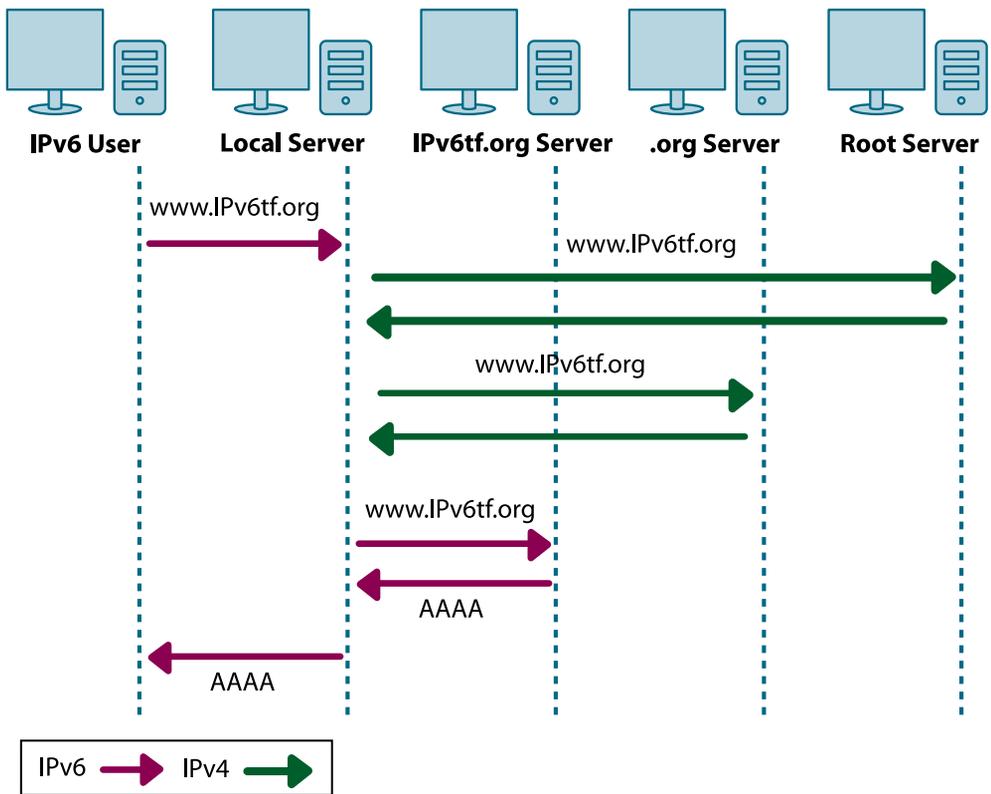
FIGURE 1: DIFFERENCE BETWEEN **DNS** TRANSPORT AND CONTENT

Because not all DNS infrastructure supports IPv6, it is currently recommended that all DNS servers be dual-stack, i.e. capable of performing DNS operations over IPv4 and IPv6. This also ensures compatibility with existing servers.

Another important concept is that of master/primary and secondary/slave servers. In short, the master server is the server in which DNS data is created and updated, that later is automatically propagated to slave servers.

9.2. Installation and configuration steps

There are several DNS server programs that support IPv6, the most common of which, both in IPv4 as well as in IPv6, are BIND for Unix-type platforms and Windows DNS Server for Windows platforms. Their installation and configuration is described below.

9.2.1 BIND

BIND (Berkeley Internet Name Domain) is the most widespread DNS server in use today and it is usually run on Linux platforms. Its configuration requires editing text files.

The installation can use each distribution's usual package management utility (apt-get, yum, up2date, rpm, etc.); alternately, the source files can be downloaded from *https://www.isc.org/software/bind* and compiled:

 # tar -xzvf bind-9.4.2-P2.tar.gz
 # cd bind-9.4.2-P2
 # ./configure
 # make
 # make install

Starting from an existing installation (BIND 9.4.2-P2) we will describe:

- How to enable answering requests over IPv6 (Listening on IPv6)

- How to associate IPv6 addresses to domain names (AAAA records)

- Reverse resolution of IPv6 addresses to domain names (PTR records)

9.2.1.1. Listening on IPv6

In our case, the main file that contains the DNS server configuration is located at /etc/named.conf. This is the file where we need to make some changes.

In order to enable listening on IPv6, the directive listen-on-v6 {}; must be added to the options section so that it is located at the beginning of the named.conf file as in the following example:

options {
    directory "/var/named/";
    listen-on-v6 { any; };
};

This will allow the DNS server to listen on all of the server's IPv6 addresses.

9.2.1.2. AAAA Records

IPv6 addresses are stored in AAAA-type records in the DNS. Every DNS server has what are known as zone files that contain the DNS information relating to a subdomain. We will use the subdomain example.com.

In BIND, the zones which will be handled by the server are configured in /etc/named.conf. For example, to load the example.com subdomain zone (contained in file /var/named/example.com.zone) when the master or primary server[1] starts, add the following lines:

---

1 To configure the server as a secondary or slave server, use type slave.

```
zone "example.com" {
    type master;
    file "example.com.zone";
};
```

Direct resolution zone files may contain records with both IPv4 and IPv6 addresses simultaneously. Continuing with our example, we must edit /var/named/example.com. zone and add the following:

```
ipv4-ipv6   IN   A      10.0.0.3
            IN   AAAA   2001:db8:1:0:0:0:1234:5678

ipv6        IN   AAAA   2001:db8:1:0:0:0:1234:5678

ipv4        IN   A      10.0.0.3
```

We have configured the following:

- ipv4.example.com resolves only to an IPv4 address (10.0.0.3).

- ipv6.example.com resolves only to an IPv6 address (2001:db8:1:0:0:0:1234:5678).

- ipv4-ipv6.example.com resolves to an IPv4 address and an IPv6 address simultaneously (the operating system and/or application will decide whether to use one address or the other).

9.2.1.3. PTR Records

This type of PTR record is not new, as it is the same one used for reverse resolution of IPv4 addresses to domain names. The difference with IPv6 is the notation used to represent IPv6 addresses (nibble notation[2]) and the domain name used for reverse resolution (IP6. ARPA). Zone files for IPv6 address reverse resolution contain only IPv6 addresses.

Now let's see an example with IPv6.

In /etc/named.conf we must configure the reverse resolution zone corresponding to prefix 2001:db8:1: :/48 which we have been delegated for our networks:

```
zone "1.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa" {
type master;
file "2001_0db8_0001.zone";
};
```

---

2 A nibble contains four bits and is therefore usually represented in hexadecimal format.

As we can see, the prefix is divided into nibbles and then linked in reverse order to the ip6.arpa domain.

File 2001_0db8_0001.zone will contain:
$TTL 86400
@ IN SOA ns1.example.com. dnsadmin.example.com (
  2002071901 ; serial
  28800 ; refresh
  7200 ; retry
  604800 ; expire
  86400 ; ttk
)
IN NS ns1.example.com.

4.3.2.1.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0 IN PTR www.example.com.
8.7.6.5.4.3.2.1.0.0.0.0.0.0.0.0.0.0.0.0 IN PTR ipv6.example.com.

Which means that:

- Address 2001:db8:1: :1000:1234 will resolve to the domain name *www.example.com*

- Address 2001:db8:1: :1234:5678 will resolve to the domain name *ipv6.example.com*

It must be remembered that each IPv6 address has a total of 32 nibbles, therefore the sum of the number of nibbles configured in named.conf (12 in our example) plus the number of nibbles in each address in the zone file (20 in our example) must be 32.

9.2.1. 4. Testing the configuration

In order to check that the changes have been applied, we must first restart the DNS server (in our example we must use /etc/init.d/named restart).

We can see how the server is listening on IPv6 and IPv4 addresses on the DNS port (port 53):

```
# netstat -tan
Proto  Recv-Q Send-Q Local Address                    Foreign Address     State
...
tcp    0      0      : :1:53                           : ::*                LISTEN
tcp    0      0      2001:db8:1:0:0:0:1234:5678:53     : ::*                LISTEN
tcp    0      0      10.0.0.3:53                       0.0.0.0:*           LISTEN
tcp    0      0      127.0.0.1:53                      0.0.0.0:*           LISTEN
...
```

From the same server we can use the dig client application, which allows us to query our server.

To resolve *ipv6.example.com*:
 # dig any ipv6.example.com

; <<>> DiG 9.4.2-P2 <<>> any ipv6.example.com
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48527
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 6

;; QUESTION SECTION:
; ipv6.example.com.   IN   ANY

;; ANSWER SECTION:
ipv6.example.com.   172800   IN   AAAA   2001:db8:1:0:0:0:1234:5678
...
;; Query time: 4 msec
;; SERVER: ::1#53(::1)
;; WHEN: Wed Jun 17 17:23:48 2009
;; MSG SIZE rcvd: 296

To resolve *ipv4-ipv6.example.com*:
 # dig any ipv4-ipv6.example.com

...
;; QUESTION SECTION:
; ipv4-ipv6.example.com.   IN   ANY

;; ANSWER SECTION:
ipv4-ipv6.example.com. 172800 IN   A   10.0.0.3
ipv4-ipv6.example.com. 172800 IN   AAAA   2001:db8:1:0:0:0:1234:5678
...

For the reverse resolution of 2001:db8::1000:1234
 # dig -x 2001:db8::1000:1234

; <<>> DiG 9.4.2-P2 <<>> -x 2001:db8::1000:1234
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 1333
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 4

```
;; QUESTION SECTION:
;4.3.2.1.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa. IN PTR

;; ANSWER SECTION:
4.3.2.1.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.  172800  IN
PTR www.example.com.

;; Query time: 1 msec
;; SERVER: : :1#53(: :1)
;; WHEN: Wed Jun 17 17:37:02 2009
;; MSG SIZE  rcvd: 270
```

9.2.1.5. Hint file (additional section)

The use of a hint file is configured in /etc/named.conf. This hint file contains the IP addresses of the DNS root servers.

```
zone "." {
    type hint;
    file "named.root";
};
```

Without going into detail, these servers have recently begun to be accessible via IPv6 (4 February 2008) and the hint file has been conveniently updated.

For this reason, in order to complete IPv6 configuration on your DNS server, you will need to update the hint file to include the necessary root server IPv6 addresses. The version that was available at http://www.internic.net/zones/named.root on December 12, 2008, showed that seven of the thirteen root servers already had a related IPv6 address.

The following is an example of the named.root file:
```
.               3600000    IN  NS       A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.    3600000    A        198.41.0.4
A.ROOT-SERVERS.NET.    3600000    AAAA 2001:503:BA3E: :2:30
```

9.2.2. Windows DNS Server

Microsoft Windows DNS Server runs on Windows servers. In this example we will use a fully updated Windows Server 2003 R2 SP2 Standard Edition.

It can be installed/uninstalled using Add or Remove Programs from the control Panel. Add/Remove Windows Components allows accessing the Windows Components Wizard.
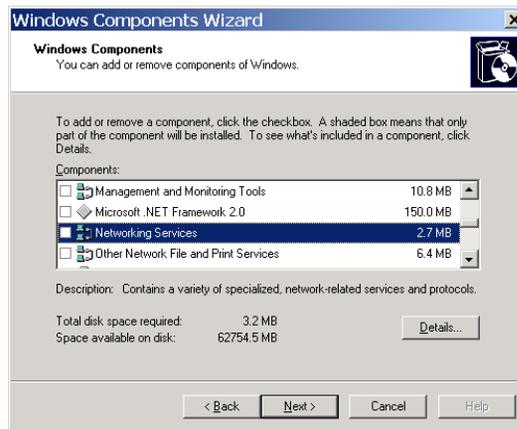
In the Windows Components Wizard, select Network Services and click on Details…
The DNS server is called Domain Name System (DNS):
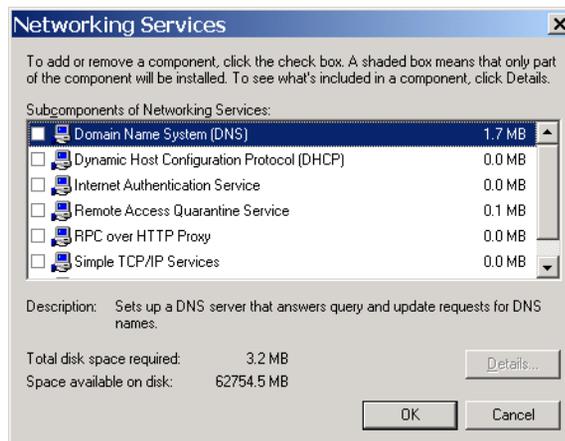
NOTE: A Windows Server 2003 installation CD will be required.

9.2.2.1. Listening on IPv6

Once the DNS server and IPv6 are installed (C:\>netsh interface ipv6 install) we need the DNS server to listen on IPv6. For this use the following:

C:\>dnscmd /config /EnableIPv6 1
Registry property EnableIPv6 successfully reset.
Command completed successfully.

NOTE: dnscmd.exe is part of the Windows Server 2003 Support Tools. It can be found in the Support\Tools folder on the Windows Server 2003 CD and installed running suptools. msi in that folder.

The DNS server must be restarted so that it begins listening on IPv6. To do this, go to Administrative Tools and run the Services Management application. Locate and restart the DNS server.

The netstat command can be used to verify that the DNS server (port 53) is listening on IPv6:

```
C:\>netstat -a -n -p udpv6
Active connections
Proto  Local address                      Remote address    Status
UDP    [: :]:53                           [: :]:0           LISTENING  0
...
UDP    [2001:db8:1: :1000:1234]:53        [: :]:0           LISTENING  0
UDP    [fe80: :1%1]:53                    [: :]:0           LISTENING  0
UDP    [fe80: :ffff:ffff:fffd%6]:53       [: :]:0           LISTENING  0
UDP    [fe80: :200:1cff:feb5:5a88%5]:53   [: :]:0           LISTENING  0
```

9.2.2.2. AAAA Records

Slave or secondary servers for a zone that contains AAAA records with IPv6 addresses can be configured using the graphic user interface. However, this is only possible if the zone's master server and remaining slave servers have accessible IPv4 addresses, as the graphic user interface does not allow entering IPv6 addresses for them.

In order to configure a new domain with a secondary or slave server, the graphic configuration interface must be used (DNS tool found under Administrative Tools).

To configure a direct resolution zone (domain name to IP address mapping), right click on Forward Lookup Zones and select New Zone. The New Zone Creation Wizard will start:

Under Zone Type select Secondary Zone, enter the name of the zone (e.g. example. com), and configure the IPv4 addresses of the primary server and other secondary servers, if any.
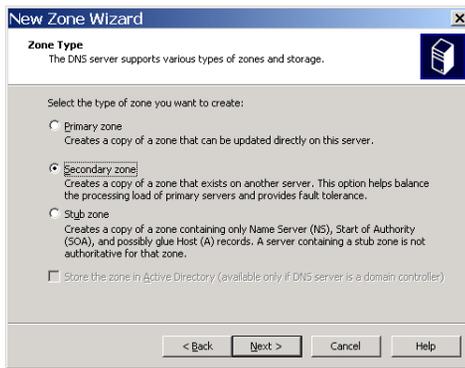
The master or primary server for a zone that contains AAAA records with IPv6 addresses can be configured using the command line user interface[3], more precisely dnscmd. The following are some of the available commands[4]:

- **Adding a zone**: dnscmd serverName /ZoneAdd zoneName zoneType [options]

- **Deleting a zone:** dnsdmd serverName /ZoneDelete zoneName [/DsDel] [/f]

- **Adding a record:** dnscmd serverName /RecordAdd zoneName nodeName [/Aging] [/OpenAcl] [Ttl] typeRR dataRR

- **Deleting a record:** dnscmd serverName /RecordDelete zoneName nodeName typeRR dataRR [/f]

- **Listing server zones:** dnscmd serverName /Enumzones

- **Viewing zone contents:** dnscmd serverName /ZonePrint zoneName
- **Listing records associated with a domain name:** dnscmd serverName> / EnumRecords <ZoneName> <NodeName>

---

3 The command line interface would also be used if it were a secondary or slave server but there was no other server accessible via IPv4.
4 Use dnscmd /? to obtain help on available commands. Use dnscmd <command> /? to obtain help on a specific command.

IPv6 Services

We will now show an example in which we will create a zone called example.com for which our server will be a primary server and where:

- *ipv4.example.com* will resolve only to an IPv4 address (10.0.0.3).

- *ipv6.example.com* will resolve only to an IPv6 address (2001:db8:1:0:0:0:1234:5678).

- *ipv4-ipv6.example.com* will resolve to an IPv4 address and an IPv6 address simultaneously (clients will decide whether to use one address or the other).

First we must create the zone using the graphic user interface. To configure a direct resolution zone (domain name to IP address mapping), right click on Forward Lookup Zones and select New Zone. The New Zone Creation Wizard will start:
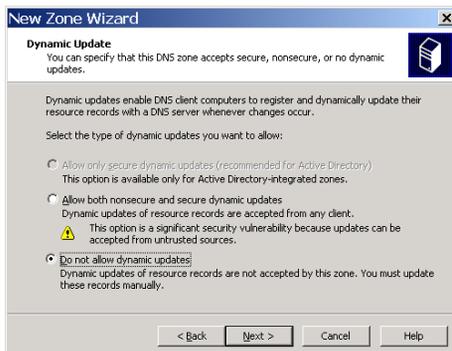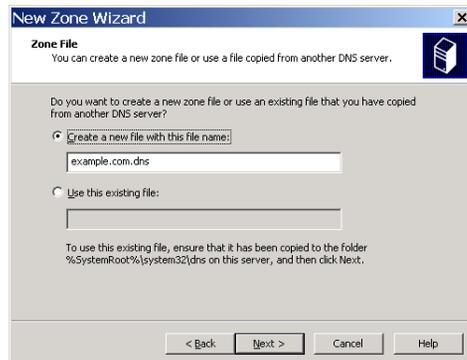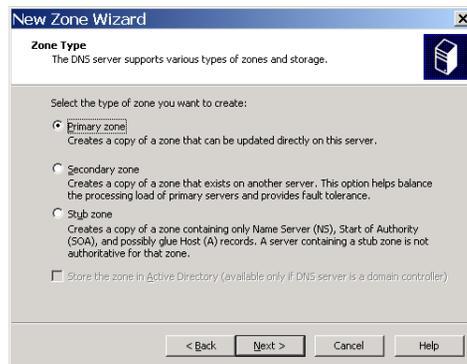
We must now enter the records using the command line:

C:\>**dnscmd : :1 /RecordAdd example.com ipv4 A** 10.0.0.3
Add A Record for ipv4.example.com at example.com
Command completed successfully.

C:\>**dnscmd : :1 /RecordAdd example.com ipv6 AAAA 2001:b8:1:0:0:0:1234:5678**
Add AAAA Record for ipv6.example.com at example.com
Command completed successfully.

C:\>dnscmd : :**1 /RecordAdd example.com ipv4-ipv6 A 10.0.0.3**
Add A Record for ipv4-ipv6.example.com at example.com
Command completed successfully.
C:\>**dnscmd::1/RecordAddexample.comipv4-ipv6AAAA2001:db8:1:0:0:0:1234:5678**

Add AAAA Record for ipv4-ipv6.example.com at example.com
Command completed successfully.

9.2.2.3. PTR Records

Slave or secondary servers for a zone that contains PTR records with domain names can be configured using the graphic user interface. However, this is only possible if the zone's master server and remaining slave servers have accessible IPv4 addresses, as the graphic user interface does not allow entering IPv6 addresses for them.

In order to configure a new domain with a secondary or slave server, the graphic configuration interface must be used (DNS tool found under Administrative Tools).

To configure a reverse resolution zone (IPv6 address to domain name mapping), right click on Reverse Lookup Zones and select New Zone. The New Zone Creation Wizard will start:

Under Zone Type select Secondary Zone, enter the name of the zone (for example 1.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa for the case of the reverse resolution of the 2001:db8:1: :/48 prefix) and configure the IPv4 addresses of the primary server and other secondary servers, if any.
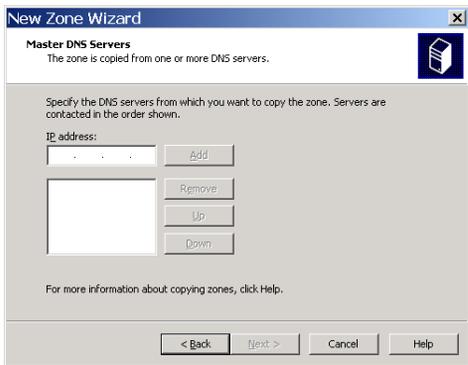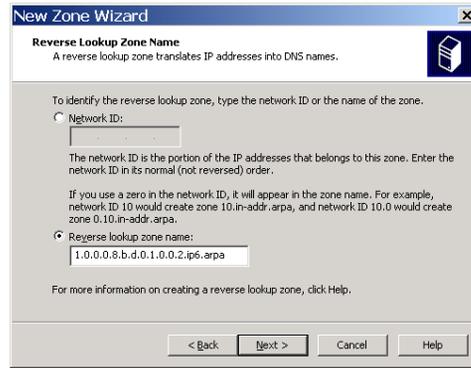
The master or primary server for a zone that contains PTR records that resolve to IPv6 addresses can be configured using the command line user interface[5], more precisely dnscmd. Further details on the most frequently used commands can be found in the preceding section.

Now let's see an example where we will create a zone e.f.e.0.8.b.d.0.1.0.0.2.ip6.arpa for which our server will be a primary server, corresponding to prefix 2001:db8:efe: /48, and where:

- 2001:db8:efe: :1000:1234 will resolve to www.example.com.

- 2001:db8:efe: :1234:5678 will resolve to ipv6.example.com.

First we must create the zone using the graphic user interface. To configure a reverse resolution zone (IPv6 address to domain name mapping), right click on Reverser Lookup Zones and select New Zone. The New Zone Creation Wizard will start:

---

5 Commands would also be used if it was a secondary or slave server and there was no other server accessible via IPv4.
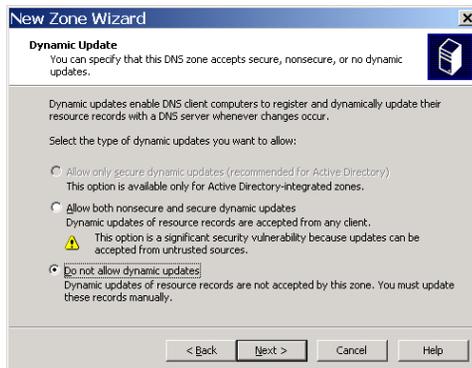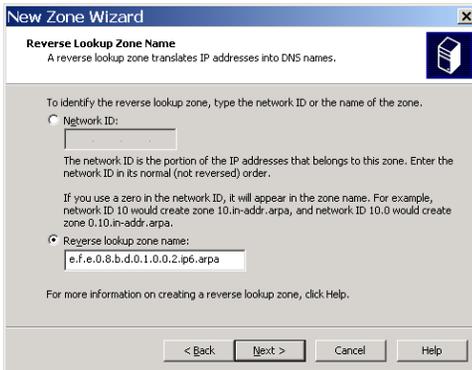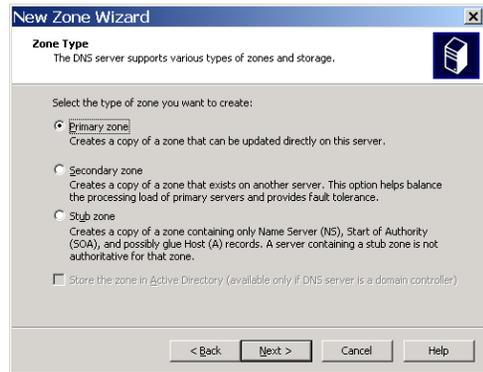
We must now enter the records using the command line:

C:\>**dnscmd : :1 /RecordAdd e.f.e.0.8.b.d.0.1.0.0.2.ip6.arpa 4.3.2.1.0.0.0.1.0.0.0.
0.0.0.0.0.0.0.0.0.0 PTR www.example.com.**

Add PTR Record for 4.3.2.1.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.e.f.e.0.8.b.d.0.1.0.0
.2.ip6.arpa at e.f.e.0.8.b.d.0.1.0.0.2.ip6.arpa

Command completed successfully.

C:\>**dnscmd : :1 /RecordAdd e.f.e.0.8.b.d.0.1.0.0.2.ip6.arpa 8.7.6.5.4.3.2.1.0.0.0.
0.0.0.0.0.0.0.0.0 PTR ipv6.example.com.**

Add PTR Record for 8.7.6.5.4.3.2.1.0.0.0.0.0.0.0.0.0.0.0.0.e.f.e.0.8.b.d.0.1.0.0
.2.ip6.arpa at e.f.e.0.8.b.d.0.1.0.0.2.ip6.arpa

Command completed successfully.

IPv6 Services

9.2.2.4. Testing the configuration

In addition to the graphic user interface that offers an easy way to visualize information, several useful commands are also available. Some of these commands are shown below for the examples completed earlier.

The following commands are used to list the direct resolution AAAA and A records that have been created:

C:\>**dnscmd : :1 /Enumrecords example.com ipv4**
Returned records:
@ 3600 A   10.0.0.3
Command completed successfully.

C:\>**dnscmd : :1 /Enumrecords example.com ipv4-ipv6**
Returned records:
@ 3600   A   10.0.0.3
 3600   AAAA   2001:db8:1: :1234:5678
Command completed successfully.

C:\>**dnscmd : :1 /Enumrecords example.com ipv6**
Returned records:
@ 3600   AAAA   2001:db8:1: :1234:5678
Command completed successfully.

To see the entire contents of the example.com zone:
C:\>**dnscmd : :1 /zonePrint example.com**
;
; Zone: example.com
; Server: : :1
; Time: Thu Jun 18 16:48:45 2009 UTC
;
@ 3600 NS   vw2003.
            3600   SOA   vw2003. hostmaster. 5 900 600 86400 3600
ipv4   3600   A   10.0.0.3
ipv4-ipv6 3600   A   10.0.0.3
            3600   AAAA 2001:db8:1: :1234:5678
ipv6   3600   AAAA 2001:db8:1: :1234:5678
;
; Finished zone: 4 nodes and 6 records in 0 seconds
;

To check that the secondary reverse resolution zone has been properly configured and list its contents:

C:\>dnscmd ::1 /Enumzones
Enumerated zone list:
    Zone count = 5

Zone name	Type	Storage	Properties
...			
1.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa	Secondary	File	Rev
...			

C:\>dnscmd ::1 /Zoneprint 1.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa
;
; Zone: 1.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa
; Server: ::1
; Time: Thu Jun 18 16:20:30 2009 UTC
;
@ 172800 NS   dns1.novagnet.com.
        172800 SOA   ns1.example.com. dnsadmin.example.com. 200906 1802 36000 7200 1814400 7200
4.3.2.1.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0 172800 PTR   www.example.com.
8.7.6.5.4.3.2.1.0.0.0.0.0.0.0.0.0.0.0.0 172800 PTR   ipv6.example.com.

To check that the primary reverse resolution zone has been properly configured and list its contents:

C:\>dnscmd ::1 /Enumzones
Enumerated zone list:
    Zone count = 3

Zone name	Type	Storage	Properties
...			
e.f.e.0.8.b.d.0.1.0.0.2.ip6.arpa	Primary	File	Rev
...			

C:\>dnscmd ::1 /Zoneprint e.f.e.0.8.b.d.0.1.0.0.2.ip6.arpa
;
; Zone: e.f.e.0.8.b.d.0.1.0.0.2.ip6.arpa
; Server: ::1
; Time: Thu Jun 18 17:09:41 2009 UTC
;
@ 3600 NS   vw2003.
        3600 SOA   vw2003. hostmaster. 3 900 600 86400 3600
4.3.2.1.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0 3600 PTR   www.example.com.
8.7.6.5.4.3.2.1.0.0.0.0.0.0.0.0.0.0.0.0 3600 PTR   ipv6.example.com.

The most common DNS client tool found in Windows environments is nslookup, a tool that is roughly equivalent to dig in Linux. Let's see some practical cases to test what we configured earlier in our examples. For direct resolution:

```
C:\>nslookup
> server 127.0.0.1
Default server:  localhost
Address:  127.0.0.1

> set type=ANY

> ipv4.example.com
ipv4.example.com        Internet address = 10.0.0.3

> ipv6.example.com
ipv6.example.com        AAAA IPv6 address = 2001:db8:1: :1234:5678

> ipv4-ipv6.example.com
ipv4-ipv6.example.com   Internet address = 10.0.0.3
ipv4-ipv6.example.com   AAAA IPv6 address = 2001:db8:1: :1234:5678
```

For reverse resolution:
```
C:\>nslookup
> server 127.0.0.1
Default server:  localhost
Address:  127.0.0.1

> set type=PTR

>4.3.2.1.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.e.f.e.0.8.b.d.0.1.0.0.2.ip6.arpa

4.3.2.1.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.e.f.e.0.8.b.d.0.1.0.0.2.ip6.arpa
name = www.example.com

>8.7.6.5.4.3.2.1.0.0.0.0.0.0.0.0.0.0.0.0.e.f.e.0.8.b.d.0.1.0.0.2.ip6.arpa

8.7.6.5.4.3.2.1.0.0.0.0.0.0.0.0.0.0.0.0.e.f.e.0.8.b.d.0.1.0.0.2.ip6.arpa
name = ipv6.example.com
```

10. Clients

Clients for the services described above are already installed by default or easy to obtain and install for practically every operating system currently in use (more specifically, their latest versions).

Operating System / Service	BSD	Linux	Mac OS X	Windows XP SP1 & later, Vista, 7, 2003, 2008
Telnet	Command line	Command line	Command line	Command line, PuTTY
SSH	Command line, OpenSSH	Command line, OpenSSH	Command line	PuTTY, SecureCRT SSH
FTP	Command line	Command line	Command line	SmartFTP
Email	Thunderbird	Thunderbird	Apple Mail, Thunderbird	Outlook
Multimedia Player	VLC	VLC	VLC, iTunes	Windows Media Player, VLC, Winamp
Web Browser	Firefox, Opera, Chrome, etc.	Firefox, Opera, Chrome, etc.	Safari, Firefox, Opera, Chrome, etc.	Internet Explorer, Firefox, Opera, Chrome, etc.
DNS	Supported	Supported	Supported	Supported

11. References

DAVIES, J. (2008). **Understanding IPv6, Second Edition**, Estados Unidos: Microsoft Press.

MALONE, D., MURPHY, N. (2005). **IPv6 Network Administration**, United States: O'Reilly

VAN BEIJNUM, I. (2006). **Running IPv6,** United States: Apress.

Apache HTTP Server Project. < **http://httpd.apache.org**> accessed June 15, 2009.

Comparison of IPv6 application support. <**http://en.wikipedia.org/wiki/Comparison_of_IPv6_application_support**> accessed June 15, 2009.

Internet Information Services. <**http://www.microsoft.com/windowsserver2003/iis/default.mspx**> accessed June 15, 2009.

IPv6 to Standard. <**http://www.ipv6-to-standard.org**> accessed June 15, 2009.

ISC BIND. <**https://www.isc.org/software/bind**> accessed June 15, 2009.

5. Enterprise Networks

1. Introduction to Enterprise Networks

Defining the exact boundary between an enterprise and a residential network is not trivial matter; both concepts are often confused, particularly since many enterprise networks use residential Internet access services. This chapter will consider as an enterprise network, one that has a clear interface with its Internet Service Provider (generally by using a firewall) and provides internal and external services.

Within the context of an enterprise network, the word addressing always brings Network Address Translation (NAT) to mind. Nearly all enterprise networks implement NAT for their IPv4 Internet access, placing a clear border between the company's internal network and the Internet. Unlike in service providers, IPv4 NAT scales well in enterprises, as it provides enough addresses for practically any known enterprise size implementation. But what is lost with the use of NATv4 (NAT for IPv4)? We have heard of the "End-to-End Principle" and how it is violated by NAT. On the other hand, in many cases, the only external service that users need in an enterprise network is web access, while the rest of the services are resolved by using internal IT services. Companies that require high levels of interaction among its users and external parties represent a special case in which the use of NATv4 may cause problems.

IPv4 exhaustion has already been discussed in the introduction, and it is clear that enterprise networks must prepare for IPv6 implementation. But if my company has enough IPv4 addresses to perform NAT, why would it need IPv6?

Answers to that question include the following:
- Users within a enterprise network may need to access content that will only be available in IPv6.

- The external services provided by the enterprise network should be reachable over IPv6, as potentially some external clients will only have IPv6 addresses.

New enterprise networks may face with an even greater challenge as they may not even have the IPv4 address needed to perform NATv4. For these new networks the IETF is working on defining translation mechanisms that will help them to transition to IPv6, particularly the NAT64 standards.

Every implementation of a new technology begins with a preliminary project, the development and implementation times of which will depend, for example, on the size of the network. Perhaps one of the greatest challenges posed by IPv6 is that understanding its true impact requires a detailed knowledge of the company's equipment and applications. Sadly, sometimes this knowledge does not exist and this makes it difficult to assess the impact of IPv6 on a fully operational installation.

This chapter begins by describing the preliminary study that must be conducted before beginning to plan the implementation of IPv6 within the company, and then goes on to discuss the different aspects that make up an implementation plan.

2. Preliminary Study before IPv6 Implementation

The preliminary study that must be performed to achieve a successful IPv6 implementation at company level include: education, impact assessment, conducting a first experience and preparing a preliminary project.

Education is crucial for conducting a proper analysis of any new technology, and IPv6 is no exception. Different information sources such as books, application and equipment manuals, standards, conferences and courses in general are available. After studying this information, network administrators should be able to tell whether or not IPv6 will affect the company's network (the answer will most probably be affirmative) and whether they have the abilities needed to study the impact of IPv6 on the company's infrastructure or should seek additional help.

After reaching the conclusion that IPv6 will indeed affect the company's infrastructure, it is necessary to analyze *where* IPv6 will impact the infrastructure, as the new protocol may not only affect the equipment but possibly also my business, whatever it may be.

The best way to begin working on those complex projects is to start an iterative process based on a specific objective. For this reason we will analyze two concrete examples: on one hand, a company that provides Internet hosting services; on the other, a small company with browser terminals.

In the case of the hosting company *(see Figure 1)*, the goal for IPv6 implementation is to make sure that all hosted contents are available over IPv6. The IPv6 impact analysis may conclude that to achieve this objective internal communications such as SQL connections, application server accesses, etc. do not require IPv6 support. This simple analysis concluded with an IPv6 implementation that only covered the network access and web front-end, which resulted in simplified tasks and lower costs.

In the second case, a company with browser terminals *(see Figure 2)* is evaluated to see whether it will be necessary to implement dual-stack in the terminals to be able to access IPv6 contents. In addition, the servers in contact with the Internet must also implement dual-stack, for example, to be able to send emails to SMTP servers that only implement IPv6.
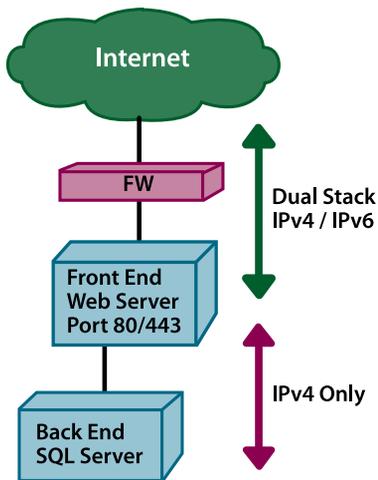
FIGURE 1: HOSTING COMPANY
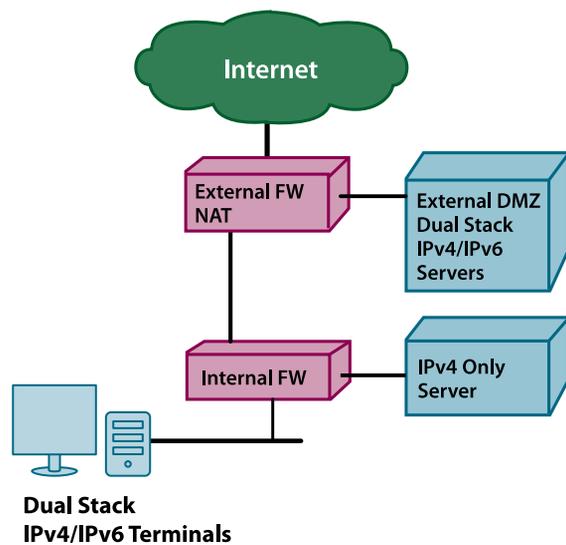
FIGURE 2: COMPANY WITH BROWSER TERMINALS

⏻ Caution! Trying to implement IPv6 in every single network device is a perfectly valid option. However, this must be a conscious decision made by the administrator. In some cases you may even think on planning to run your network only on IPv6 in order to reduce the operational extra cost of a dual stack network.

Figure 3 shows some of the elements that might be affected by the implementation of IPv6 and that administrators should consider.

Once a working objective has been established and the impact of IPv6 has been assessed, the costs involved can be estimated and we can move on to the planning stage.
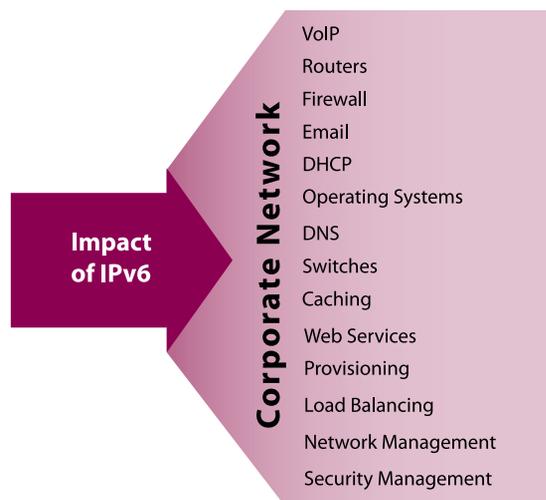
FIGURE 3: DIFFERENT ELEMENTS THAT MUST BE EVALUATED WHEN STUDYING THE IMPACT OF IPv6

Enterprise Networks

In order to properly evaluate the impact of IPv6 at the enterprise level, issues as diverse as network addressing, routing, applications, and security processes must be examined. Figure 3 shows different elements that must be considered when studying the impact of IPv6 on a enterprise network.

3. Planning IPv6 Implementation for Enterprise Networks

Planning the implementation of IPv6 involves various aspects such as:

- Addressing

- Routing

- Security

- Services

⏻ **In general, IPv6 planning results in imitating what has already been done for IPv4.**

Most enterprise networks will be dual-stack (most likely with private IPv4 addresses) and therefore IPv4 and IPv6 will coexist. However, implementing IPv6 also provides administrators with a potential new beginning and the possibility of implementng any pending infrastructure changes.

3.1. Addressing plan

The addressing plan within a company is quite simple. In general, a /64 prefix is used as a unit for every broadcast domain. For this reason, /64s are used for Local Area Networks (LANs), Wide Area Networks (WANs), and loopbacks. Alternative, longer prefix lengths are possible for Loopbacks (i.e. /128) or WANs (i.e. /126 or /127) and are sometimes preferred (see RFC 6164). Typically, regardless of its size, a company will receive a /48 from its provider – which is equivalent to 65,535 /64 networks. However, if after considering its total number of networks (LANs, WANs and loopbacks) and forecasting a 300% growth a company requires more than a /48, it should request a larger address prefix from its provider or address registry (if using provider independent addresses).

In IPv6 there are enough global unicast addresses for any company, which begs the question whether it makes senses to use NAT. Although it is up to the reader to answer this question, we will continue our study using only global unicast addresses.

We will use an example to explain how to distribute address space within a company. Figure 4 shows a typical company with a DMZ at its head office (or Office 1) and two other branch offices (Office 2 and Office 3).
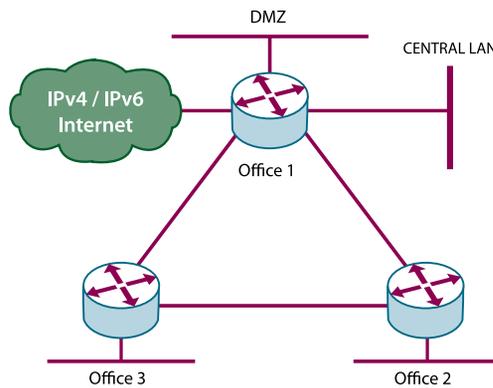
FIGURE 4: **EXAMPLE OF A ENTERPRISE NETWORK WITH A HEAD OFFICE AND TWO BRANCH OFFICES**

Whether the company obtains addresses from its connectivity provider or from its national or regional registry (for example LACNIC), the company will usually receive a /48 for its internal addressing. So, let's suppose that it receives prefix 2001:DB8: :/48 which must be divided so that it can be used to cover all of the company's networks. In IPv6 terminals in LAN´s are not longer counted, as each LAN will be assigned a /64 which will allow addressing all desired terminals. Instead, what matters is the number of networks and subnets to be addressed.

Every IPv6 address has three fields: the globally routed prefix, a subnet identifier, and an interface identifier as shown in *Figure 5*.

n bits	m bits	128-n-m bits
Global unicast prefix	**Subnet identifier**	**Interface identifier**

FIGURE 5: **CONSTRUCTING AN IPV6 ADDRESS WITH ITS THREE COMPONENTS**

In general, for a enterprise network the length of the global unicast prefix is supplied by the provider (in our case n = 48). Typically, we will also choose a 64 bits long interface identifier for two reasons: facilitating autoconfiguration in local networks and respecting the fact that often the equipment has been specifically configured to work with IPv6 addresses of that length. This means that the subnet identifier will have a length m = 16. If one wishes to perform geographic aggregation within the network (recommended to avoid the appearance of multiple networks in the internal routing table), two elements must be identified within the 16 bits corresponding to the subnet identifier: the office within the company and the network within each office.

Table 1 shows some possible ways to divide the /48 splitting **m** into different multiples of 2. The first column shows the separation of the subnet identifier, beginning with the most significant bit (to the left).

Division	Number of offices	Number of networks per office
/50	4	16,384
/52	16	4,096
/56	256	256
/58	1,024	64
/60	4,096	16
/62	16,384	4

Table 1: **Some options for dividing a /48 within a company**

In our example, let's suppose that we select /56 as an internal division because it represents the best compromise between the expected growth within each office and the growth in the number of offices. Of the 256 existing /56 prefixes we must choose one for each office, one for the external network (which we assume separate from Office 1), one for the company's WAN, and one for equipment loopbacks. To simplify aggregation in case of future growth, it is best not to utilize sequential addressing. The most efficient way to assign these prefixes is binomially, assigning from the first prefix, then from the last prefix, and then always assigning prefixes located between the two prefixes that are furthest away from each other. In this particular case they are assigned in the following order:

1- 2001:DB8: :/56
2- 2001:DB8:0:FF: :/56
3- 2001:DB8:0:7F: :/56
4- 2001:DB8:0:3F: :/56
5- 2001:DB8:0:B0: :/56
6- ...

The problem with this method is that it results in prefixes with identifiers that don't have any particular meaning and are therefore difficult to remember. Another, less efficient possibility for assigning addresses is to do it smartly, as in the following example, where we have attempted to place the office number within the corresponding address prefix. Table 2 shows the address distribution proposed for simplifying the operation.

/56 Address prefix	Destination
2001:DB8: :/56	External networks (DMZ)
2001:DB8:0:1000: :/56	Office 1.
2001:DB8:0:2000: :/56	Office 2.
2001:DB8:0:3000: :/56	Office 3.
2001:DB8:0:AA00: :/56	Where WAN networks 2001:DB8:0:AAXY: :/64 are chosen for connections between Office X and Office Y.
2001:DB8:0:BB00: :/56	Where loopbacks 2001:DB8:0:BBXX: :1/64 are chosen for Office X's router.

Table 2: **Addressing proposal for the network of the example**

The result of the addressing plan for this network is shown in Figure 6, in which all the networks to be utilized are specified.

Address assigned
by the provider

Office 1
Loop: 2001:DB8:0:BB11::1/64

IPv4 / IPv6
Internet

DMZ
2001:DB8:0::/64

LAN:
2001:DB8:0:1000::/64

WAN
2001:DB8:AA12::/64

WAN
2001:DB8:AA13::/64

Office 2
Loop: 2001:DB8:0:BB22::1/64

WAN
2001:DB8:AA23::/64

Office 3
Loop: 2001:DB8:0:BB33::1/64

LAN:2001:DB8:0:2000::/64

LAN:2001:DB8:0:3000::/64
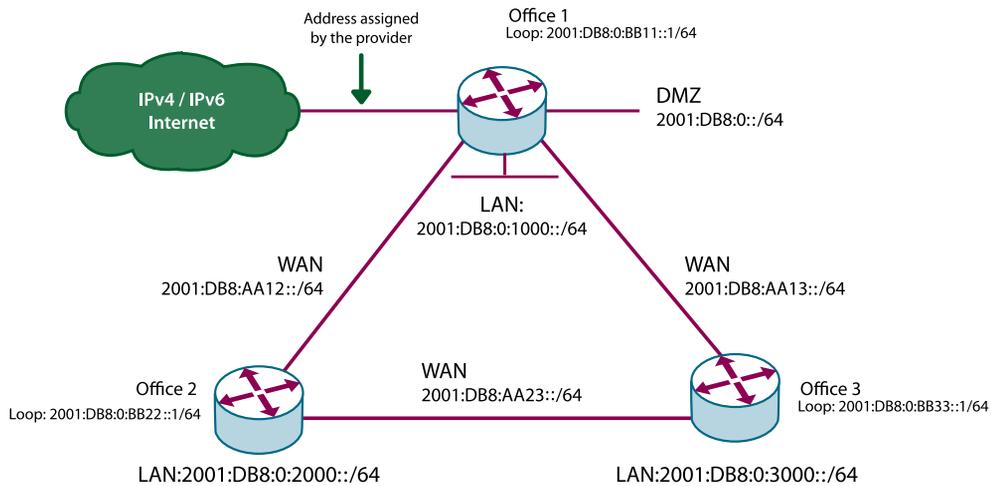
FIGURE 6: ENTERPRISE NETWORK CONSIDERED IN THE EXAMPLE

It is important to get used to the idea that some IPv6 addresses will be wasted, but this should not overly concern enterprise network designers, as the provider will usually assign the company many more addresses than it actually needs without asking any questions.

3.1.1. Server addressing

Static addressing is normally used to choose the interface identifier for each LAN corresponding to a server. This is done to try to maximize availability and avoid having to make changes when faced with network addressing problems. The static IPv6 address used for a server must be chosen based on one of the the following criteria:

- Choosing an address that is easy to remember, for example 2001:DB8::1. This option simplifies operation and maintenance because it simplifies troubleshooting.

- Choosing a random address such as 2001:DB8::ACF:2311:FFED:CAFE. Although this address complicates troubleshooting, it is believed that it will be harder to find in the face of potential port scanning attacks.

When evaluating which of the two options to choose, bear in mind that IPv6 space is very broad and that, unless DNS records are available, brute force or sequential port scanning in search of valid addresses may take a very long time. Therefore, if a server has a publicly accessible DNS record, the use of random addresses is less important.

Enterprise Networks

Example of static addressing *(ifconfig in FreeBSD):*
bge0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
    options=1b<RXCSUM,TXCSUM,VLAN_MTU,VLAN_HWTAGGING>
    inet6 fe80: :21a:64ff:fe6d:367e%bge0 prefixlen 64 scopeid 0x3
    inet 192.0.2.2 netmask 0xffffff00 broadcast 192.0.2.255
    inet6 2001:DB8: :2 prefixlen 64
    ether 00:1a:64:6d:36:7e
    media: Ethernet autoselect (100baseTX <full-duplex>)
    status: active

3.1.2. Terminal addressing

Network administrators should analyze three different terminal addressing options:

- Manual addressing: In this case each of the terminals must be manually numbered.

- Stateless or serverless automatic addressing using the route advertisement mechanism: This mechanism uses ICMPv6 packets and local multicast groups. It allows configuring the IPv6 address, the length of the prefix, and the default route. Just recently, the possibility of configuring the DNS server address using route advertisements has been standardize by the RFC 6106 but it is rarely implemented yet. Consequently, the automatic configuration of a DNS server address (or a WIN server or a SIP gateway) will need to be implemented in DHCPv6, particularly through the stateless configuration option. Because no state is maintained, the network administrator has no control over which terminals connect to the IPv6 network. Anyone with access to the shared media will be able to access the network.

- Stateful automatic addressing: In this case the IPv6 address is configured using DHCPv6 just as with IPv4. This makes it possible to define a pool of addresses or even to assign particular addresses to each terminal. Using DHCPv6 stateful configuration allows stricter access control. The default route is a parameter that DHCPv6 is still unable to provide, although research is being conducted in this sense. For this reason, even when using stateful DHCPv6, a router announcement mechanism must be used to provide the default route. Please verify support for stateful DHCPv6 in your clients' operational systems before adopting this strategy. As in 2011, not all operational system had stateful DHCPv6 support by default.

- Please verify support for stateful DHCPv6 in your clients' operational systems before adopting this strategy. As in 2011, not all operational system had stateful DHCPv6 support by default.

Regardless of the addressing mechanism that is selected, having applications that allow managing IPv6 addresses is recommended. At the date of publishing this document, some available options included Cisco Network Registrar, IP Plan (free software), IP

Address Management Module (Men & Mice), Address Commander (Incognito Software), IPAM (Blue Cat) and VitalQIP (Alcatel Lucent).

3.2. Routing plan

The IPv6 routing plan must not differ substantially from what is already being done in IPv4. It usually makes sense for a company to maintain for IPv6 the same topology used for IPv4, as maintaining two different topologies would imply increasing the operating costs of network routing as well as the number of incidents.

Two options exist for IPv6 routing:

- Static routing.

- Dynamic routing.

Specifically within dynamic IPv6 routing the following categories exist:

- Distance vector protocols: RIPNG (RIP Next Generation).

- Path vector protocols: BGPv4.

- Link status protocols: ISIS or OSPFv3.

Along with all the options listed above it is important to pay special attention to the know-how that already exists within the company. If OSPFv2 is being used for the IPv4 network it will make sense to use OSPFv3 for IPv6, just as using BGPv4 for external routing. If static routing is being used for IPv4, the same configurations can be used for IPv6.

Whenever possible the use of RIPNG should be avoided, as it has longer convergence times and presents partial topology knowledge issues. In addition, the use of RIPNG does not allow the use of modern traffic engineering techniques.

3.3. IPv6 security planning

Deploying IPv6 means that we are enabling access through a new network layer. This implies that existing IPv4 perimeter security rules are no longer valid. However, security involves not only configuring the firewall or other equipment; it also involves reviewing and analyzing processes and procedures that have been developed through the years, often following international recommendations such as ISO 27000. It is important to bear in mind that the term IP is used to designate the Internet Protocol that involves both IPv4 and IPv6. This distinction must be taken into consideration in the corresponding procedures.

Perimeter security requires configuring IPv6 rules similar to existing IPv4 rules. We will use the following example for configuring rules for a web server using IPFW in a FreeBSD device:

ipfw add 1020 permit log tcp from any to "$ip4" dst-port 80 setup keep-state in via bge0

ipfw add 2020 permit log tcp from any to "$ip6" dst-port 80 setup keep-state in via bge0

In this example, variable $ip4 represents the web server's IPv4 address and the variable $ip6 its IPv6 address. IPFW detects whether the rule should be applied to IPv4 or IPv6 depending on the format of the addresses contained in the rule. When analyzing which firewall to use, we must check whether in addition to IPv4 rules it supports IPv6 rules, as well as its ability to maintain state in IPv6.

There are two aspects that require special attention when configuring a firewall in IPv6 as opposed to IPv4: ICMPv6 and multicast.

In the case of ICMPv6, new message types must be considered, particularly type 128 which is now "echo request" and type 129 which is "echo reply". Even more important is understanding that IPv6 does not perform fragmentation in routers and that it only performs end-to-end fragmentation. The path MTU discovery process (PMTUD) is implemented to enable successful communication within an environment with different MTUs. The PMTUD process is explained in Figure 7, where the firewall must allow type 2 ICMPv6 packets to reach the server. RFC4890 contains recommendations for filtering ICMPv6 packets.

Another interesting case is that of multicast filtering, particularly link-local multicast filtering (ff02: :/16 addresses). In IPv6 there are no broadcast addresses and features such as address autoconfiguration, duplicate address detection, and neighbor discovery rely on the use of multicast. A firewall's filtering of link-local multicast will interfere with these functions and prevent their proper operation in IPv6.
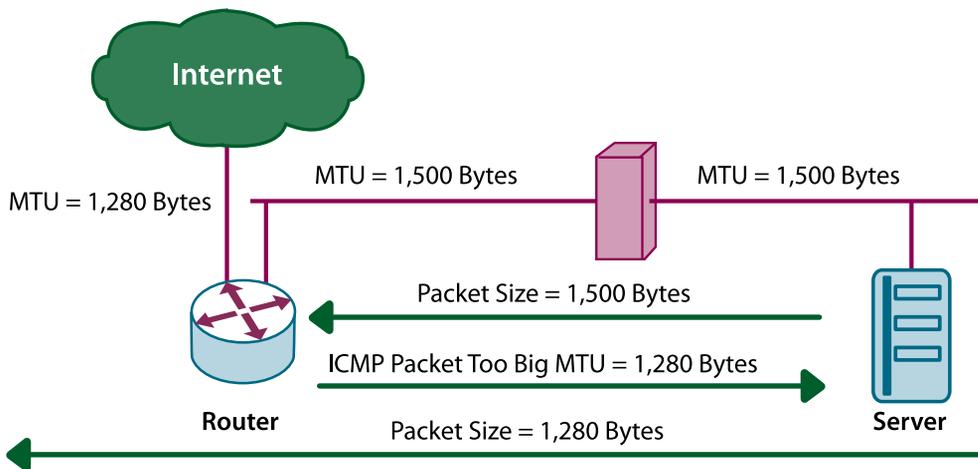

FIGURE 7: **PATH MTU DISCOVERY**

IPv6 for All. A Guide for IPv6 Usage and Application.

As shown in Figure 7, for MTU discovery the server sends a 1,500-byte packet which is discarded by some router at some point along the way because it is larger than the MTU of the next link. The router sends back to the server an ICMPv6 "Packet too Big" (type = 2) control packet containing the information of the MTU which caused the problem. After receiving this message the server adjusts the packet for the path's new MTU. If the firewall located along the path were to prefix the type = 2 ICMPv6 packets and prevent them from reaching the server, communication would not occur.

In addition to configuring the firewall, the rest of the equipment that considers perimeter security such as IDS, log analysis tools, etc. must be upgraded to support IPv6.

3.4. Service plan and IPv6

Companies should adapt their services so that they are able to support IPv6. These services can be either external or internal, and can involve commercial, open source, or custom applications. As a practical rule, any application or equipment that handles IP packets or IP addresses must be studied to understand its IPv6 support or lack thereof.

More specifically, mail services, web services, chat services, DNS services, and management systems (particularly address management systems) should be checked.

The chapter on **"IPv6 Services"** contains examples of how to configure various services so that they support IPv6.

4. Transition of a Enterprise Network to IPv6 and Depletion of IPv4 Addresses

The transition from an IPv4-only network to a network with IPv6 support will not be completed in one day as was the case when switching to digital television, or the change from NCP to TCP/IP Internet that occurred on January 1st, 1983. Instead, this process will be gradual: as most traffic sources and/or receivers transition to IPv6, more traffic will turn to the new protocol. As a result, IPv4 and IPv6 will coexist on the same physical media for many years to come and it is not even clear whether IPv4 will ever completely disappear. For enterprise network administrators this should come as no surprise, as they already have plenty of experience in the coexistence of IPv4 with other protocols such as IPX, AppleTalk or DECnet on the same physical media.

Because it is gradual, this transition is not coordinated between clients and servers. Perhaps at one point some clients will have IPv6 support while some servers won't or vice versa. To this, we must add the fact that possibly clients' operating systems include automatic tunneling mechanisms such as 6to4 or Teredo which make these clients' IPv6 connectivity less than optimum. In operating systems such as Vista the user is often unaware that this feature is enabled.

Two clearly separate elements should be considered when preparing a plan for implementing IPv6 in a enterprise network: infrastructure and services. In both cases the most important aspect to consider is service degradation perceived by the clients.

In the case of infrastructure (configuration of routers, firewalls, databases, management applications), the degradation could be due to an IPv6 connection with excessive delay. When a client starts using IPv6, after finding an IPv6 service this connection will be preferred over IPv4. If the IPv6 connection's delay is much greater than that of the IPv4 connection the client will feel its impact. To solve this problem native connections or tunneling to nearby points should be preferred. If the company has its own routing policy it should attempt to configure all its connections with IPv6.

In the case of service configuration, the critical point is the moment when IPv6 addresses are published through their AAAA records on the DNS servers. From that moment on, external and internal clients who have some sort of IPv6 access (native or through tunnels) will begin to prefer those over IPv4 access. In this regard it is important to consider that AAAA records must not be configured for services that do not have IPv6 access and that initially it may be a good practice to conduct experiments using domains of the type *ipv6.mycompany.mydomain*.

As we saw earlier, the possibility exists that in the near future public IPv4 addresses will no longer be assigned. In this scenario of IPv4 address scarcity companies will be faced with two challenges:

- The possibility of not having enough IPv4 addresses to provide their services.

- Having to provide services to clients that only have IPv6 addresses.

For the first scenario, the extreme case would be that of a company that has no IPv4 addresses at all but must access content both in IPv6 and IPv4, as well as provide services to external clients over IPv6 and IPv4. The solution to this challenge is called NAT64/DNS64, based on translations between the two Internet Protocols. Service Providers may also offer translation services to enterprise customers through a scenario called "Carrier Grade NAT or CGN.

The second scenario represents a company whose clients only have access to IPv6 and must therefore make sure that its services are reachable by these clients.

There are undoubtedly many uncertainties as to how these changes will occur in the coming years and which technological options will prevail. However, every possible scenario involves the presence of IPv6 on enterprise networks, and this means that companies must prepare for this as-yet uncertain future that will bring with it countless opportunities.

6. Research and Education Environments

1. Introduction

In this chapter we will describe IPv6 deployment in research and education networks. Those usually comprise universities and research centers but in some cases can also include schools and other related organizations. This sector's experience is worth noting, as it has led the development of the new version of the IP protocol and has the most experience in its deployment.

Throughout the chapter we will describe some of the advantages that IPv6 offers for this environment and some of the major networks around the world that already deployed this protocol. In addition, we will provide the practical information necessary for a university or research center to easily deploy IPv6 within its network.

2. Why is IPv6 Used in Education and Research?

We have already mentioned that the academic/scientific sector has led the adoption of the IPv6 protocol and is currently one of the most experienced sectors in this area. But what are the reasons behind this? This section will attempt to answer this question.

2.1. A bit of history

If we go back to the early days of the Internet we can see that its underlying technology was closely related to research and education environments. Bear in mind, for example, that the protocols that make up the foundation of today's Internet – TCP/IP – were the result of research projects conducted in the United States within the area of defense, yet Stanford University and the University College of London played a key role in their development.

The experiences that contributed to the development and evolution of the Internet as we know it today also include the protocols that made possible the World Wide Web and the HTTP[1] protocol, all of which were developed based on the needs detected through research conducted at CERN[2]. Likewise, the first graphical browser, something that seems so natural to us today, was developed in 1992 by the National Center for Supercomputing Applications, NCSA[3]. Along the same lines we could also mention most of the technologies, protocols, software, and applications currently used on the Internet.

Something similar occurs with IPv6. When the IETF took on the task of researching and promoting the creation of a new version of the IP protocol that would overcome the limitations inherent to version 4, the process was once again lead by groups based in universities and research centers such as MIT, Harvard University and CERN, among others, in cooperation with leading Internet companies (see for example RFC1752[4]).

---

1 http://www.w3.org/Protocols/HTTP/AsImplemented.html
2 http://public.web.cern.ch/public/
3 http://www.ncsa.uiuc.edu/
4 http://www.ietf.org/rfc/rfc1752.txt

2.2. Previous experiences

One of the first experiences in the use of IPv6 was conducted within the framework of the 6bone project, an attempt to set up a "virtual" network as a testing field for the new version of the protocol. We refer to this as a virtual network because many connections were initially based on tunnels encapsulated within IPv4 Internet links, although later they were transitioned to native connections. This experiment was concluded in 2006.

The first large-scale IPv6 deployments were made within the framework of academic or research networks such as Abilene (Internet2) in the United States, Geant in Europe, CERNET2 and CSTNET2 in China, and WIDE and JGN2 in Japan.

Europe has promoted the advancement of IPv6 through various research projects and initiatives such as, for example, 6NET[5], Euro6IX[6] and GEANT[7].

In addition to these initiatives, the development of IPv6 versions for BSD and Linux operating systems through the KAME and USAGI projects, respectively, deserve a special mention because universities have actively participated and played a key role in these projects.

All the references mentioned above serve to highlight the close relationship between academia and the research environments and the development and utilization of the new version of the IP protocol. This is the reason why in this sector the use of the protocol has spread with the greatest speed; it is also the reason why the sector was the first to detect the possibilities offered by the new features. Some examples of this will be mentioned in the following section.

2. 3. Services and applications used in academic networks

A peculiarity of current academic/scientific networks is that they provide services that are uncommon to other types of networks. We'll mention some examples below and see how IPv6 can help us take advantage of them.

- This type of networks currently have grids – systems found on a layer between the applications and the network services and used for sharing resources which may be globally distributed allowing access from remote locations. These resources may be computing resources or data storage capacity, the use of costly or difficult to access equipment such as high-technology microscopes or telescopes installed at remote locations.

5 http://www.6net.org/
6 http://www.euro6ix.org/
7 http://www.geant.net
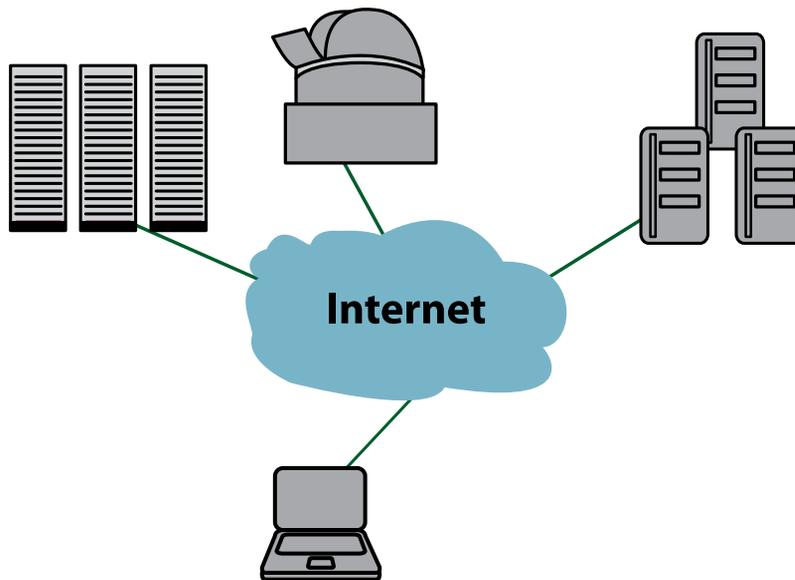
FIGURE 1: **RESOURCE SHARING ON A GRID**

This type of system allows equipment to be used by a group of people beyond the organization that owns it, which is why we speak of "virtual organizations". This chapter does not intend to provide in-depth information on grid systems, but will limit itself to mentioning the benefits that the incorporation of IPv6 could bring to this service. On the one hand, the security features provided by IPSec such as end-to-end payload authentication and encryption simplify privacy and control issues. The possibility of having global, publicly reachable IP addresses allows the large-scale deployment of grid services, both from the point of view of resources as well as from that of the devices able to use them.

- Multicast is another type of technology commonly used in this type of networks. It allows optimum bandwidth utilization when data is served to multiple recipients, as it is not necessary to replicate the transmission for each receiver. Consequently, content can be served with a signal of greater quality because bandwidth utilization does not grow in proportion to the number of receivers. Multicast is used for audio and video streaming, on-demand content, multi-point videoconferencing, etc. Although this technology is available in IPv4, it has also been part of the IPv6 protocol since the new protocol was first designed and is therefore much easier to implement.

- Videoconferences are part of the daily work of teachers and researchers; they are also often limited by the use of NAT. The possibility of having a public IP address enables end-to-end communications.

- As already mentioned when speaking about grids, the mobility included in the IPv6 protocol makes it easy to access the resources from within any organization. This is a highly desirable feature, as it is common for researchers to move from one working group to another.

115

Research and Education Environments

- Finally, the increase in bandwidth and data transfers have made it necessary to use so-called "jumbo frames" (9,000 bytes or more) to improve bandwidth utilization efficiency. IPv6 will allow improving these transfer rates even further by using "jumbograms" as networks are prepared for using this technology.

3. Research and Education Networks around the World

Before discussing IPv6 deployment at a university or research center we must place ourselves within the context of existing academic or research networks so as to be able to understand which services may be available.

The scientific and education sectors are currently connected through physical infrastructure, most of which has advanced features such as quality of service, multicast and, as mentioned above, native IPv6.

Figure 2 shows a map of national research and education networks (NRENs) around the world.

▮ Internet2	▮ TEIN2		
▮ Canarie	▮ EUMEDConnet		
▮ RedCLARA	▮ Connections to academic networks in countries not yet subscribed to Advanced Networks		
▮ GEANT2	▯ Countries not connected		
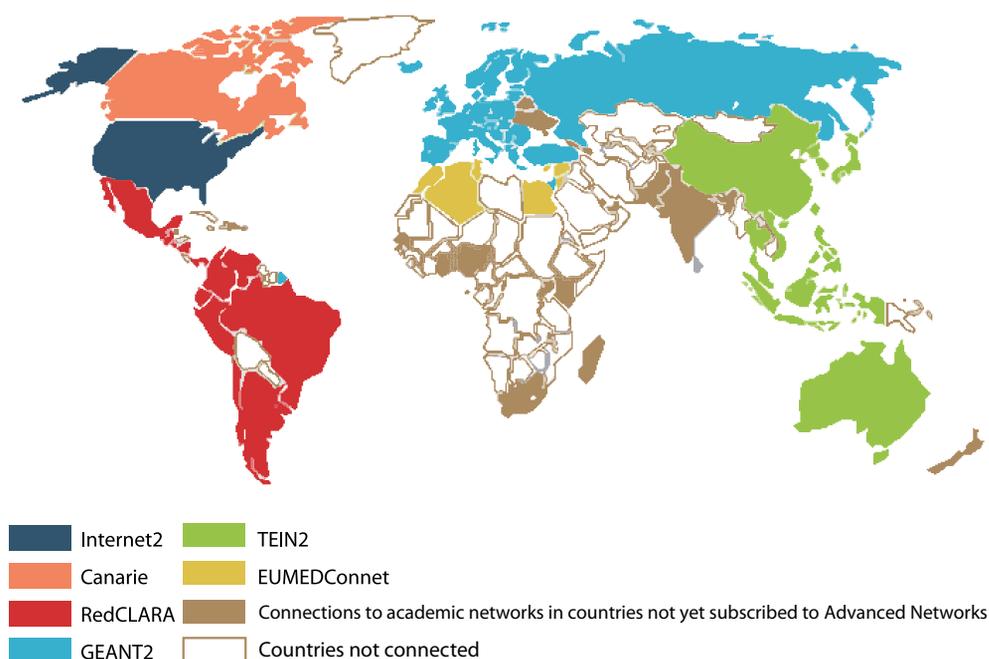
Figure 2: Map of existing NRENs[8]

Most of these networks have supported IPv6 for years, reason for which, depending on which region of the world we are located in, we will be able to take advantage of this availability to achieve native connectivity for our own institution.

8 http://www.redclara.net/index.php?option=com_wrapper&Itemid=293&lang=es

IPv6 for All. A Guide for IPv6 Usage and Application.

We will now mention the most important regional networks. It is important to highlight that many of these networks are made up by national networks (NRENs), which will ultimately provide connectivity to the end site (university, research center, institute, school, etc.). As we will see, these national networks will allow us to obtain the resources needed to implement IPv6 in our organization.

FIGURE 3: **ABILENE (INTERNET2) – UNITED STATES**[9]

9 http://www.internet2.edu/pubs/200904-Internet2CombinedInfrastructureTopology.pdf

Research and Education Environments
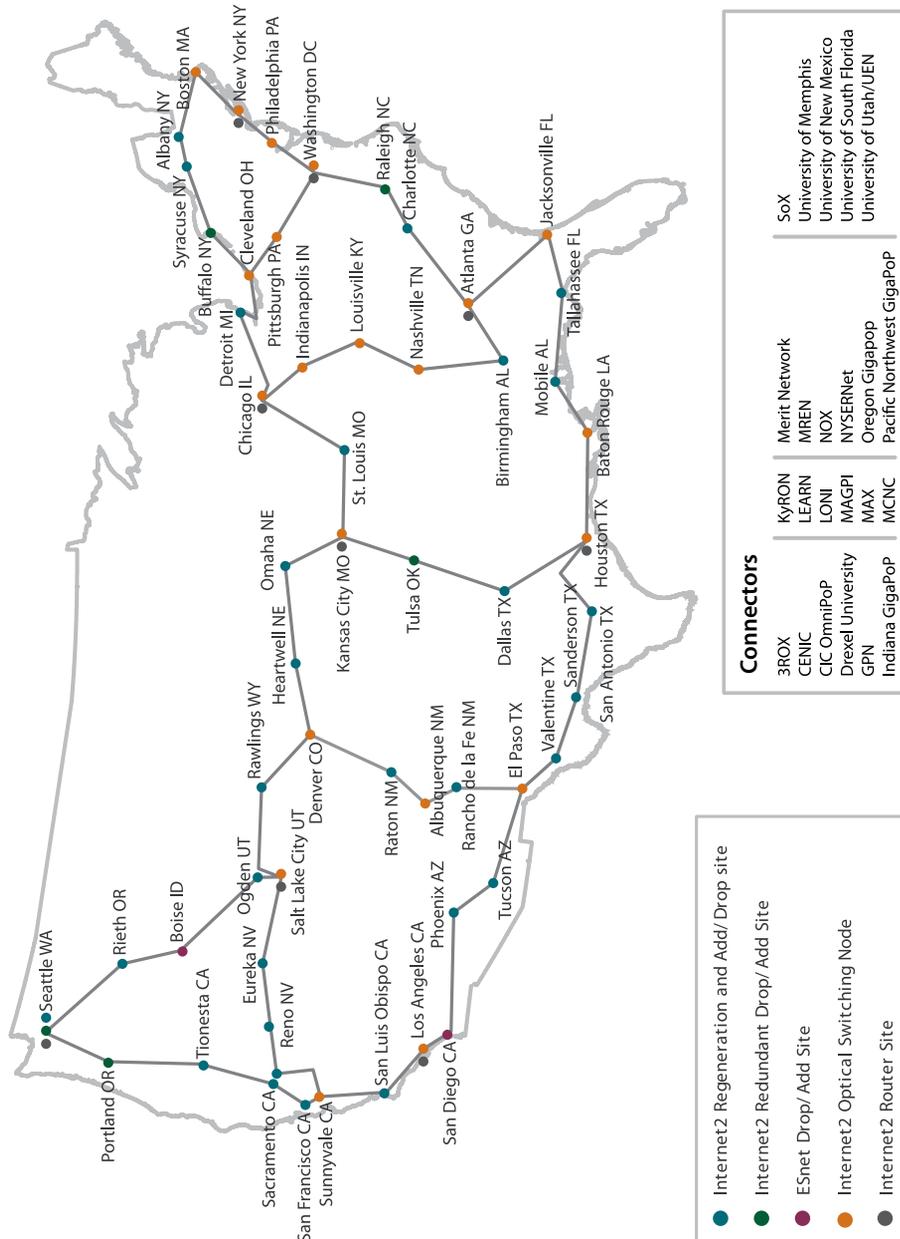
Figure 4: **RedCLARA - Latin America**[10]

10 http://www.redclara.net/index.php?option=com_content&task=view&id=51&Itemid=236

IPv6 for All. A Guide for IPv6 Usage and Application.

Legend:
- Dark fiber
- 10 Gbps.
- 2.5 Gbps.
- 310 Mbps.
- 155 Mbps.
- 34/45 Mbps.
- Planned upgrade
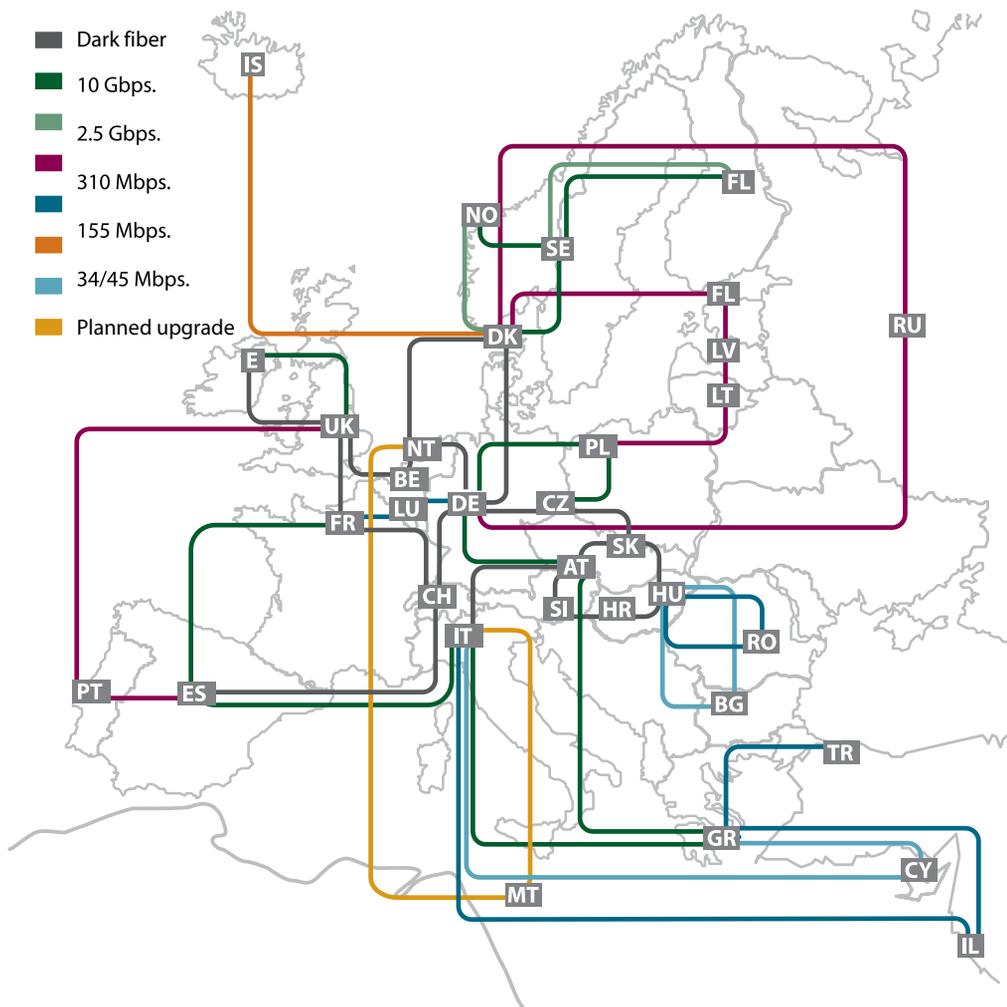
FIGURE 5: **GEANT2 – EUROPE**[11]

In the United States, universities are connected to the Abilene network through the Internet2 project as shown in Figure 3*:* Abilene (Internet2) – United States.

In Latin America, the RedCLARA network connects all national research and education networks (NRENs), see Figure 4*:* RedCLARA – Latin America.

In Europe, the GEANT network has evolved and its latest version, GEANT3, covers all European countries. It connects more than 3,500 universities and research centers. See Figure 5: GEANT2 – Europe.

_____

11 http://www.geant2.net/upload/img/jan08map.jpg

Finally, in the Asia-Pacific region, the TEIN2 network has similar characteristics and interconnects the most important national networks of that region with Europe. See Figure 6: TEIN2 – Asia-Pacific region.
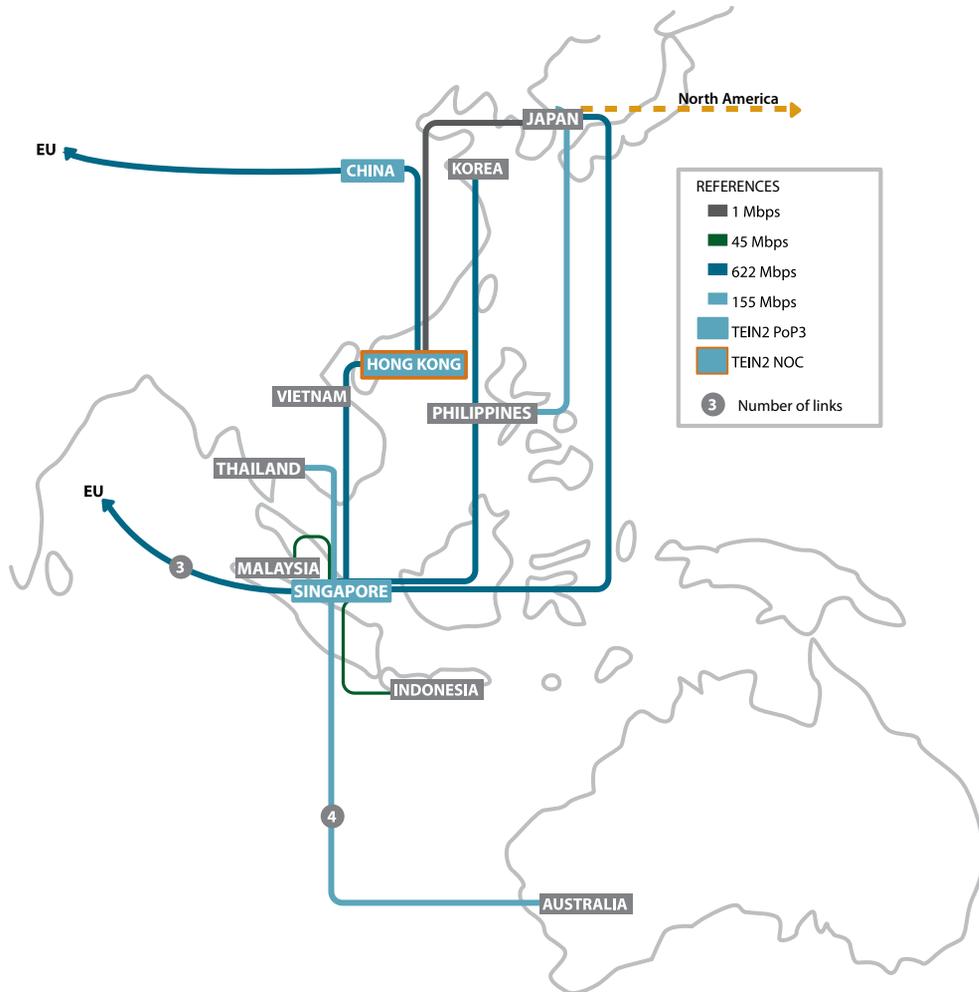
FIGURE 6: **TEIN2 - ASIA-PACIFIC REGION**[12]

4. Deploying IPv6 at Universities/Research Centers

In this section we will describe the steps needed to deploy IPv6 in a university network or, more generally speaking, in the network of an educational or scientific organization.

---

12 http://www.tein2.net/upload/img/TEIN2-web.gif

IPv6 for All. A Guide for IPv6 Usage and Application.

Although one might think that these types of organizations are no different from a enterprise or home office, their networks have certain peculiarities and, consequently, we believe that they should be dealt with in a separate section. But first it is important to clarify that we will deal with the networks used by researchers and teachers with no mention of system administration networks, as the latter are similar to those described in other chapters of this book.

4.1. Equipment, applications and services that must be considered

In general, these networks use the following equipment:

- Routers
- Servers
- Workstations (desktop and laptop computers, other devices)
- Videoconferencing equipment
- Switches (wired or wireless)
- Firewalls

Typically available services include the following:

- DNS
- Incoming and outgoing email service and mailboxes
- HTTP/HTTPS
- Directories and authentication services
- Grids
- Monitoring

Based on the above we can identify which are the most commonly used applications. We will focus on open source solutions, as they are the typical choice in the environment we are describing.

- BIND
- Sendmail or Postfix
- Apache
- OpenLDAP
- RADIUS
- Globus
- Different open source monitoring applications

The latest versions of all the applications mentioned above support IPv6, therefore all we need is to consider the corresponding configuration options. However, before discussing these options we will deal with the issue of which address ranges may be used.

4.2. How to assign IPv6 addresses at a university

When defining a range of IPv6 addresses for this type of organization we must take into consideration that, as already explained, most research centers and universities are connected to a national or regional research and education network (NREN).

Typically, the NREN already has native IPv6 connectivity and will supply a range of addresses to our organization.

In these cases a /48 prefix will be obtained, making available 256 /56 subnets for internal assignment within the organization in accordance with usual practices.

It is also important to note that academic/scientific organizations are often connected to an NREN and also obtain Internet access through an Internet service provider.
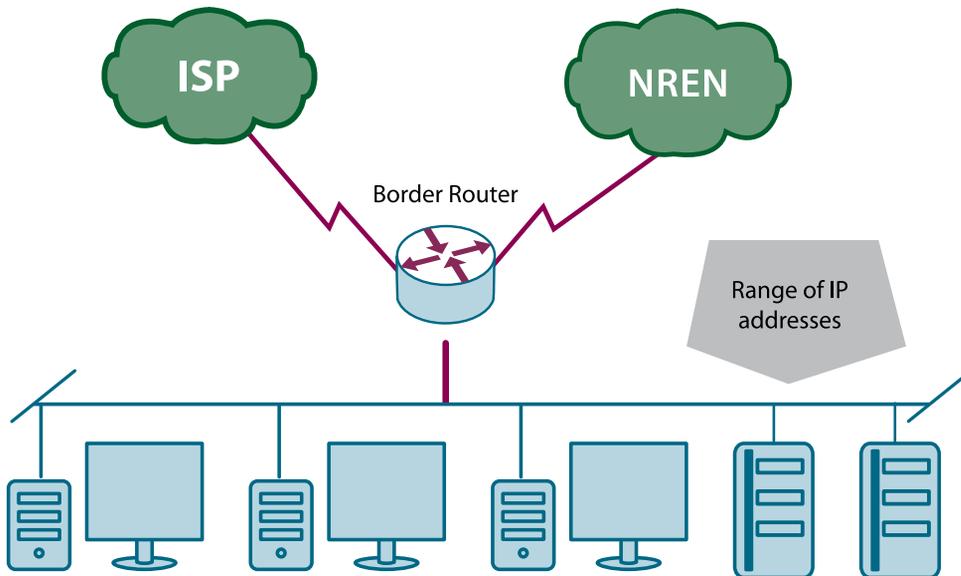
FIGURE 7: **CONNECTIVITY DIAGRAM OF A LAST MILE ORGANIZATION WITHIN AN NREN**

In this case the organizations meet the conditions required for applying for their own IPv6 address range directly from one of the Regional Internet Registries (RIRs). In addition, some of the existing RIRs have special policies that apply to organizations such as universities or research centers even if they are not multihomed.

Connections obtained by organizations that are part of an NREN deserve their own paragraph. In most cases this is implemented using a transparent point-to-point link that connects to a site where the NREN is present. In this case configuring native IPv6 will not be a problem. However, in some cases the internal connectivity of the NREN is achieved through VPN technology, usually MPLS, offered by a service provider.
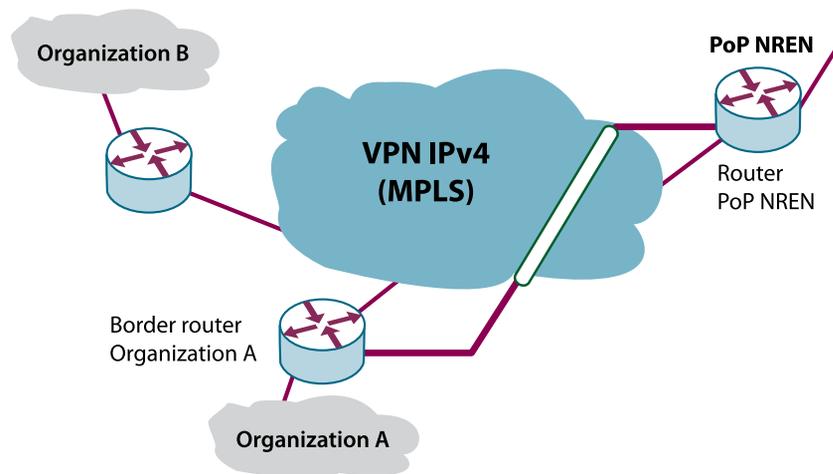
FIGURE 8: **INTERNAL CONNECTIVITY OF AN NREN THROUGH A VPN**

The question is what to do in these cases, and the solution is to use a technique for encapsulating IPv6 traffic within IPv4. This may be as simple as configuring a tunnel between the point where the NREN is present and the organization's border router. Techniques such as 6PE or carrier of carriers, among others, may also be used.

4.3. Equipment configuration

We will now briefly describe how to configure the equipment required to implement IPv6 in the network of an academic organization such as the ones mentioned earlier.

4.3.1. Routers

The organization's IPv6 prefixes must be configured in the router interfaces where IPv6 will be enabled. In order to allow the autoconfiguration of the devices it is convenient to allow network prefixes to be announced on each LAN.

A word on the internal routing protocol used by the organization: because, in addition to IPv4, it will now be necessary to include the exchange of IPv6 information, an internal routing system that supports IPv6 must be used. Therefore, we recommend using OSPFv3 or IS-IS, both of which allow handling different topologies in each version of the protocol. This is important when deploying IPv6 within the internal network in stages, as otherwise the existence of intermediate equipment that may not support IPv6 would create routing problems.

Configuration example (Cisco):
interface GigabitEthernet0/1
ipv6 address 2001:0db8:1009:101: :1/64
ipv6 nd prefix 2001:0db8:1009:101: :1/64
ipv6 ospf 1 area 0

```
ipv6 router ospf 1
router-id 1.1.1.1
area 0 range 2001:db8:1009: :/48
```

To communicate with its NREN, the organization will most likely have to establish a BGP session to publish its network prefix and learn all the prefixes that are external to the organization. Alternately, if this is the only external connection, a default route could be used and the NREN will have a static route to our prefix through that connection.

Example:
```
router bgp 64500
address-family ipv6 unicast
neighbor 2001:0db8:ffff: :2/64 remote-as xxx
network 2001:0db8:1009: :/48
```

As mentioned earlier, the connection to the NREN or some part of the university's internal network may need to traverse an IPv4-only network. In this case it will be necessary to manually set up a tunnel between the endpoints that support IPv6.

Example:
```
interface tunnel 1000
ipv6 address 2001:db8:FFFF:FFFF: :1/64
tunnel source GigabitEthernet 0/1
tunnel destination 10.1.1.1
tunnel mode ipv6ip
```

4.3.2. Servers

This type of organizations provide services using Linux or Unix operating systems. Stable versions of both these operating systems that include IPv6 support have been available for many years and therefore their configuration should not represent a problem.

Typically, autoconfiguration is not used on servers and manually specified static addresses are used instead. Linux systems allow disabling autoconfiguration through a kernel parameter: "net.ipv6.conf.*.autoconf=0".

Configuring IP addresses on the interfaces varies depending on the operating system that is being used. For example, in Solaris they are configured by defining a file /etc/hostname6.xxx, where xxx is the name of the network interface, while in operating systems such as Ubuntu and Debian they are configured using the file /etc/network/interfaces.

It is usually desirable that the servers are registered in the DNS, both in the direct as well as in the reverse zones. This will be discussed later in the section on how to configure DNS.

4.3.3. Workstations (desktop and laptop computers, other devices)

Both desktop and laptop computers use operating systems that support IPv6, either Linux or Windows. Certain workstations may run Unix systems which, as already mentioned, simplifies IPv6 deployment.

It is advisable that this type of devices be configured automatically, preferably with a DHCPv6 server, as this allows greater control over address assignment and provides additional information such as the DNS servers. This feature also makes it easier to automatically update the organization's DNS records.

In general, commonly used applications will have no problem dealing with the new version of the IP protocol. Among others, commonly used applications include email clients, Internet browsers and collaboration systems.

The equipment connected to a university's network may include different types of instruments used for data acquisition that have a network interface, such as microscopes, analyzers, controllers, and sensors in general. It is highly likely that these devices will not be prepared to handle IPv6. Limitations may also be encountered in network printers and scanners, which is why it is important to consider the need to maintain IPv4 compatibility. Consequently, it is recommended that user terminals retain the dual-stack system.

4.3.4. Videoconferencing equipment

In addition to videoconferencing software, academic and research organizations generally use specialized videoconferencing equipment. These usually comprise specialized hardware that run proprietary software. The most well-known brands currently available on the market include, for example, Polycom, Tandberg, Aethra, Sony, and LifeSize, all of which offer a large variety of models.

The level of IPv6 support varies depending on the brand and model of the devices, which is why providing specific recommendations for each case is outside the scope of this chapter. Nevertheless, it is worth noting that when evaluating the purchase of videoconferencing equipment its degree of IPv6 protocol adoption should be considered. In some cases only the SIP protocol can be used in IPv6 as H.323 communications are not supported. Some devices have limited manual configuration capabilities and only allow autoconfiguration mechanisms. The same considerations are valid for the devices known as multipoint control units or MCUs.

4.3.5. Wired or wireless switches

Layer 2 devices should not represent a problem from the point of view of IPv6, as they do not need to interpret upper layer payloads. Nevertheless, it is convenient for switches to be able to support features such as MLD snooping that allow determining which ports to send multicast traffic to depending on the multicast groups to which the devices connected to those ports have been subscribed.

As to wireless access points, universities usually use them in transparent mode, simply as access points, and therefore they should not pose any problem. It will only be necessary to consider their IPv6 support and configuration options when those devices act as routers. However, this is not usually the way in which they are used in this type of organizations, as it would not allow centralized administration.

4.3.6. Firewalls

Firewalls represent a critical element of the available network infrastructure. We must verify that they support IPv6 and allow configuring filtering rules in a manner similar to IPv4. Analyzing the different brands available on the market is outside the scope of this work, as the variety of available options is immense. Instead, we will simply mention that there are open source solutions available that can be used such as ip6tables for Linux or ip6fw for BSD.

It is important to bear in mind that the same rules must be created in IPv6 as in IPv4, otherwise different policies would exist for one version of the protocol and the other.

4.4. Implementing IPv6 services

To conclude this section we will examine some of the services that organizations of this nature must define. In the following paragraphs we will describe how to configure some of these services and the considerations that should be taken into account in each case. For the most common services further configuration information can be found in the chapter on Services.

4.4.1. DNS

Universities and research centers usually have their own IPv4 address ranges and manage their own name servers, both for the direct as well as for the reverse zones. This is similar in the case of IPv6, which is why we will now see how to configure this service.

The most common DNS application is BIND which, as already mentioned, supports IPv6. The first consideration we must bear in mind is that the name server must have IPv6 addresses configured and be reachable within the internal network both via IPv4 and via IPv6. This will allow the DNS to respond natively in IPv6.

Just as in IPv4 "A" records define the IP addresses associated with a name, in IPv6 we will use "AAAA" records, which work in a similar manner. Usually within the same zone both types of records will be defined.

Example:
```
ns1  IN       A       192.0.2.18
     IN       AAAA    2001:0db8: :18
ns2  IN       A       192.0.2.12
     IN       AAAA    2001:0db8: :12
```

On the other hand, it will also be necessary to define the reverse resolution records in the corresponding zone, which in the case of IPv6 is ip6.arpa. The records that are used are the same PTR records used in IPv4. It is important to remember that all groups of zeros that are usually omitted when writing an IPv6 address must be completed and also that each hexadecimal digit of the address must be included in the reverse representation, separated by decimal points. This complicates the manual edition of reverse zones.

Example:
$ORIGIN 0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.

8.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0   IN PTR ns1.example.org.
2.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0   IN PTR ns2.example.org.

This way we can define all the information corresponding to the servers and devices that have statically assigned IP addresses.

One of the most powerful tools provided by IPv6 is device autoconfiguration. But how can we reflect this information in the DNS without losing the dynamics that this option provides? Simply by using the dynamic update option provided in current versions of BIND. With this technique it is possible for a node or for the DHCPv6 server to aggregate the corresponding records in the direct and reverse zones. In these cases usually a key shared between the DNS server and the client is used to authorize changes. This is why we mentioned earlier that it is easier to perform these actions from a DHCPv6 server, as this server will be the agent authorized to dynamically modify the DNS zones. The configuration of this feature is outside the scope of this chapter, but the reader may obtain reliable information in BIND reference manuals or DNS documentation.

4.4.2. Incoming and outgoing email service and mailboxes

To implement an incoming and outgoing SMTP service with IPv6 all that is required is for the server to have IPv6 addresses, as the software used to provide this service is already prepared to use the new version of the protocol.

In the case of mailboxes it is necessary to use POP3 or IMAP software solutions that support IPv6. Some existing open source applications such as the one developed by Washington University[13] support IPv6.

4.4.3. Web servers

Web servers are one of the key services in this type of networks, as there are usually both centralized as well as decentralized institutional pages for different colleges, departments, divisions, etc. Various projects, student groups, and departments will also have their own pages and generally a wide variety of servers will coexist within the same network.

13 http://www.washington.edu/imap/

Fortunately, the most widely utilized server in open source systems is Apache, which is perfectly prepared to handle IPv6.

One thing we must ensure, however, is that the server is configured to respond to requests both on IPv4 as well as on IPv6. This is often achieved simply by using a single IPv6 socket and IPv4-mapped addresses (: :ffff:a.b.c.d addresses).

For this service to function properly it is necessary to define the corresponding records in the DNS for each web server installed at the organization. As we've seen, this means defining the AAAA records as well as the reverse resolution records within the ip6.arpa hierarchy.

4.4.4. Directories and authentication services

Directories are a service commonly used at academic organizations, as they allow accessing different information regarding the persons that make up the organization and sharing it with other similar organizations. Directories allow having a centralized view of each person's data, both personal and relating to their access to different resources (workstations, network, email authentication, etc.).

The most commonly used implementation for this type of directories is LDAP, and openLDAP –which supports native IPv6– is a widely adopted open source software. As mentioned when discussing web servers, dual-stack access should be allowed under both versions of the IP protocol.

RADIUS is a service related and usually linked to the directory service. This protocol allows authenticating and authorizing access to resources in a distributed manner, directing the queries to the server of the corresponding organization. Using RADIUS the academic community can implement services such as researcher or teacher roaming between different organizations while simultaneously maintaining the same level of access to resources as if they were present at their original organization.

We must verify that the software used to implement RADIUS within our organization supports IPv6. FreeRADIUS, one available open source application, is IPv6-ready. It is also important to check that the devices can connect to that server both via IPv4 as well as via IPv6, as this will be a basic service within our networks's infrastructure.

4.4.5. Grids

We've already mentioned the advantages that IPv6 may represent for grid systems. Globus Toolkit, the most recent version of the software on which those systems are based, supports IPv6, so any new applications created based on this software will be ready for the new version of the IP protocol. The only requirement is that the server on which the system runs and the applications on which the grid services are defined are prepared for the new protocol. As we've seen, they generally are.

4.4.6. Monitoring

Typically university networks make available to their users different information systems that report on the traffic that traverses the network and allow measuring parameters such as which services are used, source and destination IPs, and even the autonomous systems with the most information exchange. In turn, network administrators both at the level of the institution as well as at the level of its components (colleges, schools, etc.) need to have control over the links that make up the network infrastructure.

There are many commonly used software packages, but we will only mention some of the ones that support IPv6: MRTG, Cacti, Nagios, Ntop, and Ethereal.

5. Additional Considerations

To conclude we will highlight some of the features of IPv6 that may benefit or prove useful for university networks and for the administrators of those networks.

5.1. Availability of addresses

Universities are characterized by having large numbers of devices, with many subnets under different administrators (colleges, departments, etc). As we've seen, many of the applications used by researchers and teachers require public, globally reachable IP addresses. IPv6 makes it easier to fulfill this requirement.

5.2. Autoconfiguration

Networks with large numbers of terminal devices and decentralized administration such as the ones set up in these institutions benefit from the possibilities of device autoconfiguration, particularly with DHCPv6. This allows maintaining an additional control over IP address assignment to wired and wireless terminal devices.

5.3. Renumbering

The possibility of easily renumbering networks simply by configuring changes in the corresponding routers is essential for large-scale networks such as the ones we are analyzing. This provides great flexibility when hiring Internet providers, as changing from one to another does not represent a major effort for network administrators.

5.4. Mobility

As already mentioned, mobility is a very common feature among researchers and teachers. For this reason, the possibility of accessing resources from different networks as if accessing them from their original workstations makes it easier for inter-institutional working groups to cooperate.

5.5. Other practical issues

We will conclude this section by mentioning some applications which, although of a more traditional nature, are also widely used in research environments.

It is common to transfer data using ftp or to execute processes in a remote device through an ssh session. These two applications allow using computing resources at other locations and transferring data and results. Both applications support IPv6 and can therefore be used without requiring any prior configuration.

Remote viewing is another common activity in these environments, either through X-windows systems or through less complex interfaces such as VNC. IPv6 can also be used for this purpose, as X-windows supports the new protocol and versions of VNC are available that contemplate IPv6.

Also common among distance learning systems are the software packages known as "virtual campuses" or "e-learning environments". One of the most commonly used applications is Moodle, which has already incorporated IPv6 support.

Finally, it is often important to know whether we have end-to-end IPv6 connectivity with a particular site, in which case we can use tools such as traceroute6 or mtr that report the path followed by our traffic before reaching its destination. If the entire path supports IPv6 we can take advantage of this availability. We can also use ping6 to check if a host or server supports IPv6.

6. Conclusions

In this chapter we've seen the particular characteristics of the networks of scientific or academic organizations. Those are the result of historical reasons relating to research on the technologies that form the basis on which the Internet is built; they are also reflected in their willingness to experiment with new applications and protocols.

We've also seen that many of the services currently required for educational or scientific purposes are different from those used in other types of networks such as enterprise networks or those of Internet providers, and for this reason they deserve a special analysis.

As stated in this chapter, most of these services and applications already support IPv6 and, combined with the existence of vast experience in the use of the new version of the protocol, we should be ready to incorporate our institutions to the networks that already have native IPv6 support.

7. Internet Service Providers (ISPs)

1. Who Should Read this Chapter?

This chapter is intended for those responsible for planning and/or operating an IP backbone and currently providing services exclusively over IPv4. This situation is typical of different types of Internet Service Providers (dedicated access, broadband providers, regional providers, etc.). To simplify configuration tasks, we will present examples that readers may use as a reference.

We will describe the issues administrators will be faced with when preparing a project. They will have to create an addressing plan, assign addresses to interfaces (certain considerations are required when using IPv6), take into consideration the services used for operating the IP backbone, deal with routing considerations when implementing IPv6 peers, etc.

The following diagram shows the typical topology used by Internet service providers and the area for which we will describe IPv6 implementation.
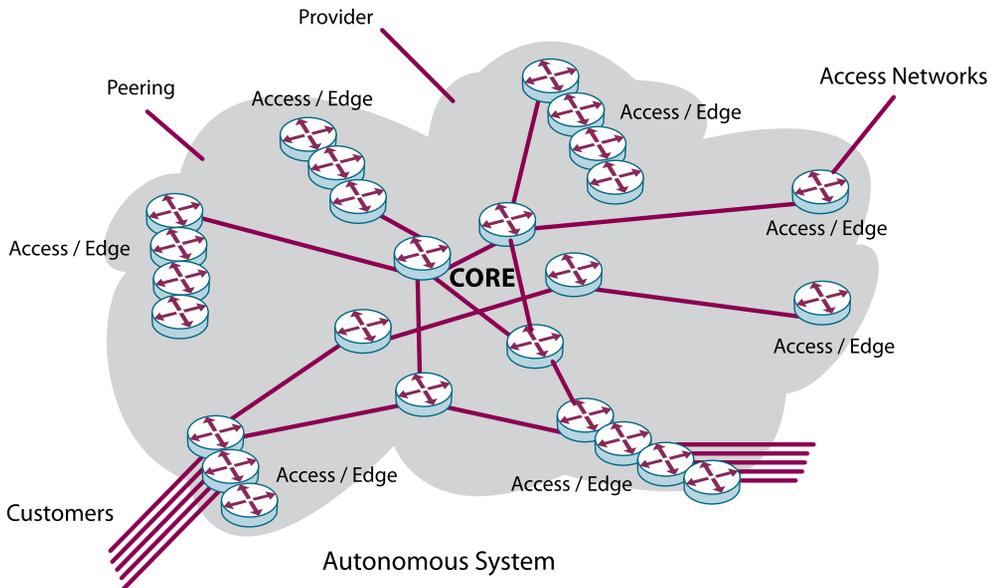
FIGURE 1: EXAMPLE OF A BACKBONE

When a provider's network has a large number of routers on its backbone, a configuration with route reflectors is used. This is necessary in order to be able to scale BGP implementation. The configuration described for these cases will serve as an example to complement BGP configurations currently in use.

We will describe the changes that must be made in the devices located within the highlighted area. This chapter does not describe what is going on in other devices, as

this has already been covered in other chapters of this book. We will not describe the configuration for customer premises equipment (CPE), local area networks (LANs), client or server hosts, or access networks used by broadband providers.

1.1. Technologies and providers covered in this chapter

The configuration examples presented in this chapter will consider both Cisco IOS as well as Juniper JunOS. Some other existing router providers have not been included; however, their use within the backbone is less common and the command line interface (CLI) of other devices is usually similar to that of Cisco IOS.

These configurations can also be used as an example for the case that both providers are being used in the backbone. Very few situations require specific configurations for Cisco/Juniper compatibility.

1.2. Description of the services for which IPv6 must be enabled

Several alternatives exist for deploying IPv6 in an IP backbone. Administrators must choose the solution best suited to their needs taking into consideration the services offered by the provider, the size of the network, and the capabilities of the installed equipment.

As in other environments, the dual-stack solution is usually the most advisable. However, at the network core, less complex methods may be used that can be implemented more quickly than dual-stack and that are not as inefficient as tunnels. Such is the case of MPLS using 6PE.

For reasons of clarity, we will present only the following two cases: dual-stack implementation and the case of an MPLS backbone. We will also mention the use of tunnels and their configuration, although this solution is not recommended. The use of 6PE does not only apply to providers who have already implemented MPLS for offering other services, but also to cases where the network core cannot support dual-stack.

1.3. Customers already using IPv6 without their provider having enabled the new protocol

Before presenting configuration examples and recommendations for the proper implementation of IPv6, it is important to bear in mind that many of your customers may already be using IPv6 without the provider having enabled the new protocol. Transition mechanisms for using IPv6 when the provider does not support the new protocol were described in previous chapters of this book. These appear to be a great advantage and might lead service providers to believe that they can postpone IPv6 implementation, as their current customers will not pressure them to do so; however, these mechanisms are not sufficient to cover every need and in many cases may complicate troubleshooting operations.

It might be useful to know how your existing customers are using these transition mechanisms. To understand this you will need to analyze which Teredo or 6to4 relay your customers are currently using. IPv6 services provided using automatic transition mechanisms may not perform adequately. In addition, transition mechanisms always present some inefficiencies, not only because of the overhead they add to each packet, but also because of the additional hops that the packets must cover between both ends and the relays.

Once IPv6 is implemented in the backbone, it is also advisable to enable some of these relays so that customers that can not connect through native IPv6 may have a better service using the 6to4 or Teredo relay installed on the provider's own network.

2. Service Components

2.1. An Internet Service Provider's network

The configurations we will describe will illustrate how to implement an IGP and BGP to enable IPv6 in the backbone. We will assume that the IGP is used exclusively to maintain the backbone's interface routing information and that BGP contains the full table and all of the provider's own networks/routes. We will only show the configurations required in OSPF and ISIS, which are the internal routing protocols most commonly in use today.

We will assume that the backbone currently has a proper routing configuration for IPv4 both at BGP as well as at IGP level, and we will describe only the additional steps needed to enable IPv6. We will not enter into the details of the traditional configurations and explanations relating to these routing protocols, as they are already well-known and widely utilized.

In the case of BGP, we will show the configurations required for implementing a full mesh of BGP neighbors that will also serve as a reference for the more general case in which route reflectors are used. The latter solution is recommended where the backbone has more than four routers.

2.1.1. Core and edge equipment

According to best practices for IP backbone design, our network will have central routers – typically known as core routers – and access devices – typically known as edge routers. For reasons of simplicity we will not include every IGP and BGP routing configuration for both cases because they are very similar. However, in cases such as MPLS, this difference is important as the modifications will only affect edge devices.

The size of the network does not have a major impact on required configurations. As we will see below, enabling IPv6 simply requires modifying the current configuration of the interfaces, BGP and, in some cases, configuring the IGP. The aim is to follow the same

standards currently used for the backbone. It is possible that the backbone uses different configurations, such as a route reflector hierarchy, adjustments to the IGP or BGP timers, etc. All of these features will also be available for IPv6.

2.1.2. Upstream providers

Before proceeding to the configuration of the network itself it is advisable to contact current providers. Nowadays many providers offer their clients native IPv6, while others offer IPv6 through tunneling mechanisms and yet others are not yet IPv6-ready. If your provider does not offer IPv6 service or can/will not include IPv4 and IPv6 service over the same port, you will have to use a tunnel. Initially, it is recommended that you contact your current provider's upstream provider or check with your local IXP whether any free IPv6 tunneling services exist in your country or region.

If no such local service exists and upstream providers do not support IPv6, a public tunneling service will have to be used. It is important to verify the quality of the IPv4 connection between your network and that on the other end of the tunnel, selecting one that does not present packet loss or excessive delay issues.

One example of a public IPv6 tunneling service is OCCAID[1], a network that uses voluntary resources to offer IPv6 transit. This service will only be available for those organizations that are able to prove that they have requested IPv6 transit from their providers and that these providers are unable to offer the service.

2.1.3. Servers and services

Services and servers do not initially require upgrading. Enabling IPv6 routing in the backbone preserves all IPv4 functionalities and no services will be affected. All we must do is verify that the operating systems in which IPv6 is enabled do not use autoconfiguration and that they begin using the new protocol. It is advisable to have the routing properly configured before beginning to work on services, servers and firewalls.

It is also important to note that, although no services will be affected, if you want your implementation to be useful it is advisable to enable IPv6 in public servers. There will not be much IPv6 traffic if our DNS servers only reply to IPv4 queries. The reader will find all the details required for configuring any servers and services in the corresponding chapter.

2.1.4. Peerings

Currently a large proportion of IPv6 traffic may be obtained through peerings and free interconnections. Typically, providers are more flexible when accepting a free traffic interconnection using IPv6 than using IPv4. Our recommendation is to first contact current IPv4 peers to verify whether they support IPv6 interconnections and, if so, under which terms.

---

1 http://www.occaid.org/initiatives.php?node=gips

It is common for local IXPs to have IPv6 projects. In some cases these projects only comprise peering among members; however, in other cases they include services such as transit. We recommend contacting the nearest IXP in order to gain a better understanding of the implementation status in your city or country.

3. Implementing IPv6 in the Network

3.1. Implementation plan and recommended stages

As in the case of any other network project, detailed planning is required before making any changes to the equipment. Every one of the configurations described below will have an impact on the IP backbone and may affect the service.

In general, the first stage for deploying IPv6 in the backbone consists of training the staff that will be in charge of the project. Being aware and having a deep understanding of the services, equipment and configurations of the current network is crucial for making the proper decisions during the planning stage. It is also important to be in contact with the current network equipment provider to know their limitations and have their proper support from the very beginning of the process.

In parallel to this, IPv6 addresses may be requested from the regional registry. Current policy and procedure details will be described later in other sections of this chapter. The addressing plan can be prepared assuming that a /32 (if this is appropriate to the size of the network) will be received, without having to wait for the actual prefix. Once the registry has informed which prefix will be assigned, updating the spreadsheet will be extremely simple. We ust need to change the first 8 hexadecimal digits of the prefix.

Note that we are referring to a spreadsheet, however, due to the size of the IPv6 addressing space is very common to use IPAM (IP Address Management) tools or devices.

It is then advisable to analyze different alternatives for the implementation (dual-stack, 6PE, tunneling, etc.) and to decide which will be the most appropriate considering the services already configured on the network. This chapter will allow readers to identify existing alternatives and the configurations they involve, and to decide which one is best suited to a particular network.

Once the devices that will require changes are identified (depending on the type of implementation and current routing configurations), it will be necessary to survey available resources (available memory, processor utilization, etc.) so as to guarantee that the configurations will not compromise the equipment's proper operation. It is important to remember that the topology used for IPv6 does not necessarily have to be the same as the one used for IPv4. If a link or device does not support IPv6 but other paths are available, the IGP can be configured to see only valid paths. If an IS-IS IGP is used, implementing this alternative will be somewhat more complex.

Now we are ready to decide the configuration plan, first enabling IPv6 on routers, then configuring the IGP if necessary and, finally, enabling IPv6 in BGP sessions.

4. Receiving IPv6 Prefixes from the Regional Internet Registry

Because this chapter is aimed at Internet service providers, we recommend using IPv6 prefixes received directly from the corresponding Regional Internet Registry or RIR. Possibly, in the case of IPv4 addresses, the provider did not qualify to receive the prefixes directly from the RIR. Nevertheless, it is advisable to check the registry's policies once again. For example, a broadband provider that is using private IPv4 addresses (RFC1918) for its clients and wishes to replace those private addresses with public IPv6 addresses (in this case global IPv6 addresses) may present an application to the registry requesting the IPv6 addresses needed to complete this change.

We will now describe some of the policies that apply to Internet service providers that wish to receive IP addresses from their registry.

Minimum allocation:
RIRs will apply a minimum size for IPv6 allocations, to facilitate prefix-based filtering. The minimum size for an allocation of IPv6 address space is a /32 (the provider may obviously apply for a larger prefix if necessary).

Consideration of IPv4 infrastructure:
When an existing IPv4 service provider requests IPv6 space for the final transition of existing services to IPv6, the number of current IPv4 customers may be used to justify a larger request than would be justified if based solely on IPv6 infrastructure.

In some cases providers do not assign IP address prefixes to their customers but have access to prefixes assigned by the registry based on End User policies. Those holding portable IPv4 addresses can obtain portable IPv6 addresses. In the case of LACNIC, the current policy states the following:

LACNIC will assign portable IPv6 address prefixes directly to end sites if they hold portable IPv4 addresses previously assigned by LACNIC.

Assignments will be made in prefixes smaller than or equal to a /32 but always greater than or equal to a /48.

In certain cases it is possible to obtain portable IPv6 addresses without previously having been assigned portable IPv4 addresses. In the case of LACNIC, the policy that allows receiving IPv6 addresses from the registry establishes the following requirements:

- In case of announcing the assignment on the Internet inter-domain routing system, the receiving organization shall announce a single prefix, aggregating the total IPv6 address assignment received.
- Provide detailed information showing how the requested prefix will be used within the following three, six and twelve months.

- Submit addressing plans for at least a year, and host numbers on each subnet.

- Submit a detailed description of the network topology.

- Prepare a detailed description of the network routing plans, including the routing protocols to be used as well as any existing limitations.

- Assignments will be made in prefixes smaller than or equal to a /32 but always greater than or equal to a /48.

5. Addressing Plan

Before presenting recommendations for the addressing plan, it is important to keep in mind the following rules:

In no case a segment or prefix may be smaller than a /64. The only exception to this rule may be the case of the interfaces; however, we will see in other sections of this chapter that using a /64 is also advisable for point-to-point links.

For assignments to customers, RFC3177 and the RIRs recommend the following criteria:

- Typically /48, except for very large subscribers – in other words, even residential users should receive a /48.

- /64 when it is known that one and only one subnet is required by design – only in the case of individual connections (dial-up connections are a good example of this).

The recommendation is to assign customers prefixes equal to or larger than a /48. Considering that the customer will not be able to divide this prefix into prefixes smaller than a /64, we will be providing customers 2^{16} (65,535) segments or prefixes for use in their internal networks, where each of these segments may have 2^{64} devices. If prefixes larger than a /48 are assigned this must be properly documented in order to justify new IPv6 prefix requests from the registry. Customers can receive a prefix larger than a /48 if they have multiple facilities or offices and each one receives a /48.

ARIN's wiki[2] contains excellent reference material on IPv6 adressing plans and is maintained and updated by the community.

---

2 http://www.getipv6.info/index.php/IPv6_Addressing_Plans.

Internet Service Providers (ISPs)

RFC4291 is also an excellent reference that describes different types of addresses, their representation, and recommendations for routers and hosts.

Each LAN (Ethernet) segment will use a /64.

It is advisable to reserve a /48 per PoP for network infrastructure. For most PoPs utilization of this prefix will likely be quite low; however, it is always recommended to have a completely independent prefix (a prefix not used for customers) with IP addresses available for each PoP in the future.

It is advisable to use a dedicated /64 for loopbacks. Considering that each loopback will use a /128, the number of addresses available in a single /64 will be sufficient.

Using a /64 for each point-to-point interface is recommended. This may appear to be a huge waste of IP addresses as a /127 might be enough (just as in the case of IPv4 we may use /31s); nevertheless the use of /127s could result in operational issues as described in RFC3627.

5.1. Policies relating to IPv6 assignments (to customers as well as internal)

Unlike the case of IPv4, there is no maximum size limitation for IPv6 assignments to customers. Each provider can have their own assignment policy to encourage optimum utilization of the total address prefix. However, all /48 assignments to end users must be registered at the RIR/NIR in order to allow proper evaluation of utilization when requesting subsequent allocations.

RIRs/NIRs are not concerned about which address size an LIR/ISP actually assigns. Consequently, RIRs/NIRs will usually not require detailed information regarding IPv6 user networks as they did with IPv4. This information will only be required in certain cases.

5.1.1. Assignment to the operator's infrastructure

Each PoP on the network may have a dedicated /48. This too may seem like an inefficient use of IPv6 addresses, yet it is in compliance with current policies. In the case of LACNIC, the corresponding policy states the following:

An organization (ISP/LIR) may assign a /48 per PoP as the service infrastructure of an IPv6 service operator. Each assignment to a PoP is regarded as one assignment regardless of the number of users using the PoP. A separate assignment can be obtained for the in-house operations of the operator.

5.2. NAT and network protection

Often the IPv4 addressing plan used for the network includes segments with private addresses (RCF1918) the object of which is to hide those devices. In those cases, Network Address Translation (NAT) is used to access the Internet from those segments. It is advisable to analyze the use of private IPs and NAT from two different points of view:

- As an "enabler" of a large number of IP addresses

- As a security resource to avoid access to protected segments

Unfortunately, the use of NAT introduces important complications in different applications and in some cases makes them impossible to use. Diagnosing connectivity problems becomes extremely complex and usually developers need to consider NAT support when they require connections between applications traversing a public network.

When IPv6 is used there is no need for NAT; in fact IPv6 NAT is not standardized. There will be sufficient IPv6 addresses for every required device. Each device will be able to have a unique IPv6 address.

As to the security benefits and the use of NAT as a means for protection, we recommend reading RFC4864, more specifically the local network protection policies that will allow achieving similar or higher levels of security without using NAT.

If an organization requires IPv6 addresses that are unique yet not globally accessible (ULA[3] or Unique Local Addresses) certain tools may be used[4].

The lack of planning and proper evaluation when implementing IPv6 may result in serious security issues. Unfortunately, ever since the 90s many users and providers have been hearing that IPv4 addresses are running out. Because so far they have not been faced with this problem, many believe that it will never happen and are therefore at risk of performing a hasty, last-minute implementation. Some IPv6 features are very convenient from a security point of view (e.g., mandatory IPSec, elimination of NAT, etc.); however, the new protocol also has a huge impact on applications and firewalls and this must be taken into consideration (e.g., the need to allow certain ICMP packets). RFC4942 presents an analysis of the impact of IPv6 implementation on security.

---

3 RFC4193: http://www.ietf.org/rfc/rfc4193.txt
4 http://www.sixxs.net/tools/grh/ula/

5.3. Configurations

5.3.1. Enabling IPv6 on routers

IPv6 is already enabled in Juniper routers. In the case of Cisco IOS, the following global commands must be used:

```
ipv6 unicast-routing
ipv6 multicast-routing
ipv6 cef
!
```

To enable IPv6 on a Juniper router interface:

```
interfaces fe-0/0/1 {
    unit 0 {
        family inet6 {
            address 2001:DB8:C003:1001: :1/64;
        }
    }
}
```

To assign an IPv6 address to a Cisco IOS interface:

```
interface GigabitEthernet1/1
description Backbone Interface
ipv6 address 2001:DB8:C003:1001: :1/64
```

5.3.2. IGP configuration

If dual-stack will be implemented throughout the backbone it is advisable to use the same IGP for both IPv6 and IPv4. If MPLS is enabled in the backbone, it will not be necessary to configure IPv6 support on the IGP and the core, as the forwarding information will be handled by the LDP or RSVP-TE.

If you are currently using OSPF for IPv4 or have plans for changing the IGP in the future, it is possible to have different IGP processes for IPv4 and IPv6. RFC4029 describes the different alternatives:

- OSPFv2 for IPv4, IS-IS for IPv6

- OSPFv2 for IPv4 and OSPFv3 for IPv6

It is also possible to use IS-IS for IPv4 and OSPFv3 for IPv6; however, this requires knowledge of and experience with both protocols. That said, it is pointless to needlessly complicate network operation.

If IS-IS is being used as IGP for IPv4 you can enable IPv6 by applying the same process.

In the case of OSPF a new process will be required.

To configure OSPF for a backbone interface on a Cisco IOS:
interface Interface-Name
description Backbone Interface
ipv6 address 2001:DB8:C003:1001: :1/64
ipv6 ospf network point-to-point
ipv6 ospf 1 area 0

To configure the process:
ipv6 router ospf 1
auto-cost reference-bandwidth 10000
router-id IP-Address
area 0 range 2001:db8:C003: :/48

5.4. BGP sessions

5.4.1. Important considerations

The BGP configurations included below are simply examples that may serve to understand the basic or most common commands required for IPv6 implementation. These commands will be added to existing BGP configurations and should not affect current sessions in any way. It is highly likely that the current BGP configurations for IPv4 will include other adjustments that may also apply to BGP sessions for IPv6 (e.g., timer modifications, limits imposed on the number of prefixes, etc.).

If, in order to ensure the announcement of IPv4 prefixes, static routes are used for the large aggregated prefix, then it is advisable to do the same for the IPv6 prefix.

For example:
In Cisco IOS:
ipv6 route 2001:DB8: :0/32 Null0 254

In Juniper:
```
routing-options {
 rib inet6.0 {
  static {
    route 2001:DB8: :0/32 {
    discard;
    install;
    readvertise;
    }
  }
 }
}
```

Internet Service Providers (ISPs)

In general, BGP requirements and operation will the same for IPv4 and IPv6. Consequently it is advisable to follow the same practices used for the current BGP configuration. This will simplify troubleshooting, as all operating areas will more easily understand the new configurations.

BGP configuration on Cisco IOS
```
router bgp ASNnumber
 address-family ipv6
 redistribute commands

 neighbor RR-IP-address activate
 neighbor RR-IP-address send-community
 neighbor RR-IP-address peer-group group-name

exit-address-family
```

Route reflector configuration in Juniper:
```
protocols {
  bgp {
    group name {
      family inet6 {
        labeled-unicast {
          explicit-null;
        }
      }
    }
  }
}
```

Route reflector configuration in Cisco IOS:
```
router bgp ASNnumber
 address-family ipv6
 neighbor IP-address activate
 neighbor IP-address route-reflector-client
 neighbor IP-address send-community
 neighbor IP-address peer-group group-name
exit-address-family
```

5.4.2. Filters

It is very likely that current BGP sessions will have filters configured to avoid the exchange of private addressing prefixes (RFC1918) or that of invalid addresses. For IPv6 BGP sessions, equivalent filters would be as follows:

In Juniper:
```
policy-options {
  policy-statement invalid-ipv6-addresses {
    term deny-IPv6 {
      from {
        route-filter 0000: :/3 orlonger
        route-filter 4000: :/2 orlonger
        route-filter 8000: :/1 orlonger
        route-filter 2001:DB8: :/32 orlonger
      }
      then {
        reject;
      }
    }
  }
}
```

In Cisco IOS:
```
ipv6 prefix-list invalid-ipv6-prefixes seq 20 permit : :/3 le 128
ipv6 prefix-list invalid-ipv6-prefixes seq 30 permit 4000: :/2 le 128
ipv6 prefix-list invalid-ipv6-prefixes seq 40 permit 8000: :/1 le 128
ipv6 prefix-list invalid-ipv6-prefixes seq 50 permit 2001:DB8: :/32 le 128
```

5.4.3. Providers using MPLS

Providers already using MPLS in their backbone can use the technique known as 6PE which allows enabling IPv6 only on provider edge (PE) devices. 6PE works similarly to MPLS VPN services. The MPLS backbone will hold information regarding the prefixes exchanged over BGPv4 and forwarding information learned through LDP or RSVP-TE. The backbone will use this information to forward the packets to the other end of the autonomous system. The intermediate equipment's forwarding information will already be known, which is why dual-stack IPv6 should be configured on all edge devices.

For in-depth details on how 6PE works the reader can check RFC4798. This RFC not only describes this technique but also explains how to use it for more complex cases such as the interconnection of different autonomous systems (using an analogy to the ASN interconnection technique used for VPNs).

The advantage of 6PE is that it is very easy to configure. Once an MPLS has been implemented, the most difficult part is solved and enabling IPv6 is easier than using VPNs. The disadvantage of 6PE is that traffic routing depends on the LSPs, not on the IP address carried by each packet. Although the same is true for IPv4 traffic, if an LSP cannot be established and an (Internet) IPv4 packet inadvertently arrives at one of the backbone's routers (including its core routers) without a label it may be incorrectly routed. In the case of IPv6 traffic this problem would cause those packets to be discarded.

Although it is also possible, the use of IPv6 on MPLS VPNs is outside the scope of this chapter because this book focuses on Internet services. RFC4364, which superseded RFC2547 (commonly referenced for MPLS VPNs), covers both IPv4 and IPv6.

The configuration included in the example describes the case of 6PE using BGP with route reflectors, as this will be the most common situation for providers that have deployed MPLS.

6PE configurations for Cisco IOS

```
mpls ipv6 source-interface Loopback0

router bgp ASNnumber
  address-family ipv6
  redistribute connected route-map connected-actions
  redistribute static route-map static-actions
  neighbor RR-IP-address activate
  neighbor RR-IP-address send-community
  neighbor RR-IP-address send-label
  neighbor RR-IP-address peer-group group-name

exit-address-family
```

6PE configurations in Juniper

```
protocols {
  bgp {
    group Name {
      family inet6 {
        labeled-unicast {
          explicit-null;
        }
      }
    }
  }
}
```

5.4.4. Use of tunnels

Using tunnels in the backbone is the least efficient solution, as all it does is provide a short-term solution that cannot be scaled. Offering IPv6 services using tunnels within the provider's network complicates troubleshooting, affects the availability of IPv6 services, and makes it difficult to make proper use of the resources (links and routers). In this case the tunnels must be configured manually. This solution should be considered temporary, not the final implementation for the network.

Should it not be possible to implement IPv6 in some part of the network, configuring a tunnel will allow hiding or creating a bridge between both IPv6 ends without modifying intermediate devices. This will result in two different network topologies, one for IPv4 and one for IPv6 traffic. The following diagram shows how the previously described network topology is modified using tunnels. Although there may be two paths available for IPv4 traffic to get from one end to another, configuring a tunnel for IPv6 traffic effectively joins them by using a single path – the tunnel – that requires that the routers at both ends be available so that traffic can traverse it.
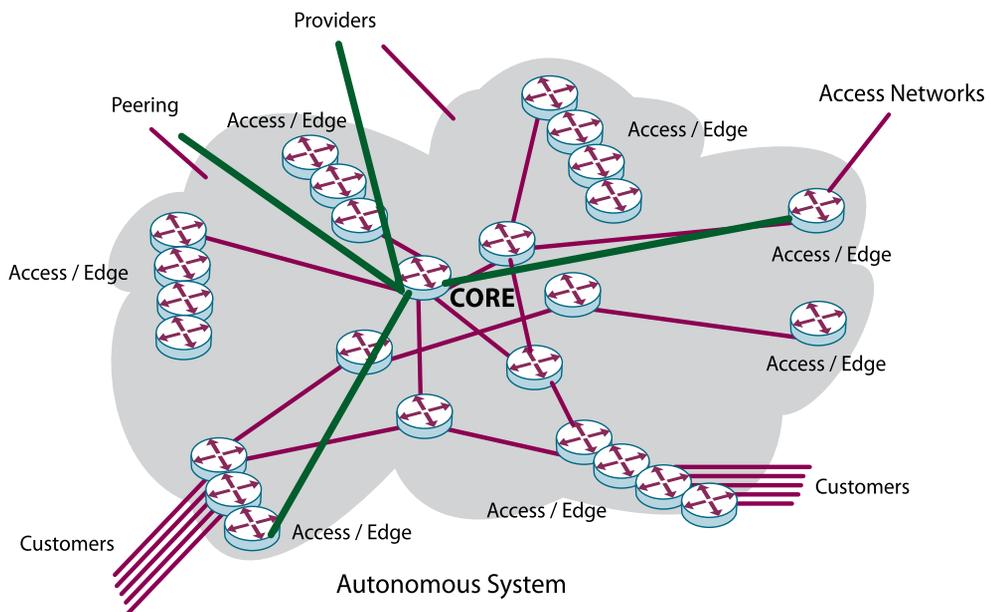

FIGURE 2: **EXAMPLE OF THE USE OF TUNNELS**

An important consideration when implementing tunnels (not only tunnels for transporting IPv6 traffic but also any other type of tunnels) is the maximum MTU supported between the ends and the maximum MTU configured in the tunnel interface. After all tunnels have been configured it is advisable to check the proper delivery of the traffic using packets of different sizes. Section 3.2 of RFC4213 contains a detailed explanation of this issue and related recommendations.

5.4.5. Tunnel choices

To separate it from other automatic tunneling techniques, the use of manual tunnels described in Section 3 of RFC4213 is called "configured tunneling". This RFC describes automatic mechanisms that are used as transitions but that are not useful in these cases. Tunnels are used within a provider's backbone to encapsulate any IPv6 packet between two end-points that can only communicate via IPv4. Once the packet reaches the final end of the tunnel using IPv4, the router recovers the encapsulated IPv6 packet and continues normal IPv6 routing.

Internet Service Providers (ISPs)

GRE tunnels (RFC2893) are the simplest mechanism available, as well as one most commonly used by service providers. These tunnels have been used for many years to encapsulate different protocols and have proven interoperability. Another possibility would be to use L2TPv3 tunnels (RFC3931).

5.4.6. Customer connections using tunnels

In the next few years it will be common for customers to request testing of IPv6 services before finally enabling dual-stack on their devices. It is also possible that your customers may need to use IPv6 to reach a router different from the one that currently connects them to the Internet. In those cases it is advisable for the provider to offer GRE tunnel terminations with IPv6 encapsulation. This will allow customers to make an initial deployment using only the router(s) they wish to involve initially.

The configuration of the tunnels is similar for all cases. The following examples are included for reference.

Cisco IOS configuration
```
interface ExampleTunnelR1
 no ip address
 ipv6 address 2001:DB8:FFFF: :17/64
 tunnel Origin-Interface
 tunnel destination Destination-IPv4-Address
 tunnel mode ipv6ip
```

Juniper configuration
```
interfaces {
   Origin-interface {
     unit UNIT {
       tunnel {
         source Origin-IPv4-Address ;
         destination Destination-IPv4-Address ;
       }
       family inet6 {
         address 2001:DB8: :17/64 ;
       }
     }
   }
}
```

6. Conclusions

Enabling IPv6 in a provider's backbone is not a complex task and will most likely not require investing in equipment. However, this task requires planning and great care at the time of implementing the necessary changes because routing configurations impacts core equipment. Fortunately, there is still time to carry out a well-planned and timely deployment.

In this chapter we have only described how to enable IPv6 routing. Offering IPv6 services may represent a more complicated task for providers, as this affects many areas not covered in this book such as network monitoring, service provisioning systems, management tools, etc. For further information on servers and services, usually the next step for Internet providers, readers can check the corresponding chapter.

8. Epilogue

An overview of IPv6 deployment around the world

Throughout the preceding chapters we discussed the current status of IPv4 addresses, what IPv6 is, why IPv6 is necessary, as well as the steps that have been taken towards IPv6 adoption in Latin America and the Caribbean, the region where this document was created.

We also explained different technical aspects relating to IPv6 deployment in different operating systems and network environments, from residential users to Internet Service Provider networks.

Current status of IPv6 deployment around the world

In very general terms ita can be said that IPv6 is indeed being deployed around the world, though in uneven steps, at different speeds, without any drastic changes occurring. In more specific terms, we should analyze what is going on in the different network environments where the new protocol is being deployed.

In the case of academic networks, deployment in Japan, Europe and North America has been very significant, mainly thanks to major public investments aimed at promoting the adoption of the new protocol. In addition, particularly in the case of Europe, together with the private sector, the European Commission has co-funded a large number of research and development projects which in turn have allowed the industry and other stakeholders to acquire the necessary knowledge to help complete IPv6 development and standardization while achieving a degree of maturity that has allowed its deployment.

As a direct result, many countries and regions have adopted public policies to highlight the fact that IPv6 deployment is not expensive if properly planned, i.e., with a certain degree of anticipation that depends on the specific case of each network, therefore ensuring that any equipment, application or service that is acquired is IPv6-ready so that when the decision is made to enable IPv6 it will not be necessary to make any new purchases.

In fact, as a result of this type of policies, several countries and regions around the world have set concrete dates as from which enabling IPv6 in public administration networks and other related networks (education, defense, etc.) will be mandatory.

From the point of view of large networks (major operators or "carriers"), particularly in the case of international operators that typically use intercontinental networks, for several years they have been making great progress and in many cases already provide comprehensive IPv6 support.

However, the situation is very different in the last mile and even in many national and regional Internet Service Provider networks. Of course, there are certain noticeable exceptions to this statement, mainly in Japan and other Asian countries, as well as a few in Europe and North America.

Status of IPv6 support in operating systems, applications and services

As you may have discovered in the preceding chapters of this book, operating systems used by computers, cellular phones and many other devices began providing IPv6 support in 2001. Today platforms that do not support the new IP protocol are relatively difficult to find. In addition, because of how IPv6 has been designed and thanks to the technical decision to allow IPv6 to be deployed in parallel with IPv4 (a technique known as co-existence), the existence of automatic transition mechanisms allows those devices to use IPv6 end to end, even if service providers do not provide IPv6 support.

The use of automatic transition mechanisms is obviously not the best solution. Instead, the ideal solution would be for Internet Service Providers to deploy IPv6 in the last mile which, as we've seen, is precisely the least developed area in all geographical regions and therefore the one that requires the greatest effort.

From an applications point of view, because operating systems generally have proper IPv6 support, it is becoming more and more common for applications to work indistinctly with IPv4 or IPv6.

As to services such as web servers, in general IPv6 adoption is moving along slowly because data centers and Internet Service Providers themselves have not yet been faced with the need to deploy the new IP protocol (e.g., immediate financial incentives). As always, exceptions exist. Among these we can mention the case of Google, a company that has made major progress in the past few years and which will undoubtedly push its competitors in the same direction.

Finally, from a traffic point of view, it is important to highlight that the situation is very different from what might be expected in view of the low level of IPv6 deployment in the last mile. Because IPv6 has been enabled in almost all user platforms (and as we've seen many operating systems are IPv6-enabled by default), and thanks to automatic transition mechanisms and some applications (mostly peer-to-peer), for more than two years IPv6

traffic that uses those transition mechanisms has been growing at a very significant rate. The most notable examples include BitTorrent, messaging applications, and even automatic virtual private network applications.

This automatic traffic increase will undoubtedly represent an important financial incentive and motivator, particularly in those regions where bandwidth is more expensive, for Internet Service Providers to deploy IPv6 in the last mile as soon as possible, or use transition mechanisms installed in their own networks (for example 6to4 and Teredo) as a temporary measure until full dual-stack can be deployed. This will allow them to save bandwidth and, indirectly, to improve the quality of the services they provide their users.

Epilogue

The IPv4 addresses used by devices to connect to the Internet are nearing exhaustion.

Combined with constant technological advances, the Internet's success and the emergence of more and newer services have resulted in the need to develop a new version of the IP protocol (IPv6) that will allow using as many IP addresses as necessary (3.4 x 10^{38} addresses).

That said, the deployment and final adoption of this new protocol must be a gradual but constant process.

For many years international organizations such as the Internet Society, LACNIC and 6DEPLOY, a project co-funded by the European Commission, have been working with great commitment in the different regions and continue to do so providing tutorials, conferences, training events, workshops, etc.

The book **"IPv6 for All"** – a project led by the ISOC Argentina Chapter – attempts to provide, in a clear and simple language, the necessary tools that will allow users to understand, deploy, and implement IPv6 in different environments.

Mónica Abalo Laforgia
President
Internet Society Argentina Chapter – ISOC-Ar

In collaboration with:

Funded by: