



Forschungszentrum
Telekommunikation
Wien

Project N4: IPv6 Transition

D1.1: General Evaluation of Transition Scenarios

Version: 1.0

Date: 15.12.2004

Authors: *I. Miladinovic (ftw.)*
K. Umschaden (IBK)

Co-Authors: *M. Banfield (TA)*
W. Bauer (KCC)
P. Tschulik (Siemens)

Status: *Final*

Document Modification Sheet

Nr.	Version	Date	Changed by	Modification
1	0.1	31.08.2004	I. Miladinovic	Creation
2	0.2	03.09.2004	I. Miladinovic, K. Umschaden	First draft of content finished
3	0.3	14.09.2004	I. Miladinovic, K. Umschaden, M. Banfield, W. Bauer, P. Tschulik	Content finished
4	0.4	30.09.2004	I. Miladinovic, K. Umschaden	Draft version finished
5	0.5	06.10.2004	I. Miladinovic, K. Umschaden, M. Banfield, W. Bauer, P. Tschulik	Review version finished
6	0.51	28.10.2004	I. Miladinovic, K. Umschaden	Backbone transition described in more detail, Cisco terminology used
7	0.6	12.11.2004	I. Miladinovic, K. Umschaden	Teredo and L2 MPLS tunneling added
8	1.0	15.12.2004	I. Miladinovic, K. Umschaden, M. Banfield, W. Bauer, P. Tschulik	Final version finished

Acknowledgments

We would like to thank the members of the N4 project for their contribution and productive discussions that helped us to write this document. Furthermore, we acknowledge the assistance of Austrian IPv6 Task Force (Transition working group) for the collaboration. Special thanks to Kurt Bauer from Vienna University for very useful input.

Contents

1	Introduction	5
2	Motivation	5
3	Existing Transition Mechanisms	7
3.1	Dual Stack	8
3.2	Tunneling	9
3.2.1	Configured IP tunnels	11
3.2.2	Tunnel Broker	11
3.2.3	6to4	12
3.2.4	Teredo	14
3.3	Translation	15
3.3.1	Stateless IP/ICMP Translation	16
3.3.2	Application Level Gateway	16
3.3.3	Network Address Translation - Protocol Translation	17
3.3.4	Transport Relay Translator	18
3.3.5	SOCKS64	18
3.4	MPLS	18
3.4.1	Native IPv6 over MPLS	19
3.4.2	IPv6 over IPv4 MPLS Core	19
3.4.3	Layer 2 tunneling over MPLS	20
3.5	Other Translation Mechanisms	21
3.5.1	ATM	21
3.5.2	Separate IPv6 Network	21
4	Transition Stages	21
4.1	Definition of Stages	23
4.2	Backbone Transition	24
4.3	Provider Access Transition	25
4.3.1	Unmanaged Customer Networks	26
4.3.2	Managed Customer Networks	26
4.3.3	General Issues for Provider Access Transition	27
4.4	Examples	27
5	Technical Issues	28
5.1	DNS Transition	28
5.2	Routing	29
5.3	Security	31
5.4	Network and Service Operation	31
6	Summary and Conclusion	33

1 Introduction

Internet Protocol version 4 (IPv4) [36] is currently the most used network layer protocol in the Internet. It has been developed in 1981 by the Internet Engineering Task Force (IETF) to cover the needs of a small backbone network linking dozens of open research and government networks. Today, there are over 60 millions nodes in the global Internet, and IPv4 is still in use.

Because of the rapid growth of the Internet, the IPv4 is running out of the available address space. Given that the exhaustion of the IPv4 address space has been long anticipated, several techniques had been developed to overcome the address shortage. They include Network Address Translation (NAT), Dynamic Host Configuration Protocol (DHCP), and Classless Inter-Domain Routing (CIDR). On the other hand, these techniques cannot meet requirements of peer-to-peer applications and the Internet's end-to-end architecture. DNS also causes problems with some application layer protocols, including H.323 and the Session Initiation Protocol (SIP), that carries the IP address in their messages. In parallel to developing these techniques, IETF began searching for a permanent solution of IPv4 address space shortage and proposed several alternatives. Among all the proposed alternatives, IP version 6 (IPv6) [11], originally called IP next generation (IPng), has become accepted. IPv6 has a 128-bit address format and offers significantly more address space than IPv4, which has a 32-bit address format.

In this deliverable, we consider different scenarios for IPv4 to IPv6 transition from the point of view of a large ISP network. The goal is to evaluate the suitability of existing transition techniques for an ISP network. This work will be presented in the context of the Austrian IPv6 Task Force industry initiative and it will give a contribution to the Austrian *roadmap to IPv6*.

This document is organized as follows. The next section states about the motivation for this work. It describes advantages of IPv6 and gives an overview of IPv4-IPv6 transition phases. Section 3 describes the most important existing transition mechanisms. It describes not only mechanisms that provide connectivity between peers that support the same IP version, such as dual stack and tunneling, but also translation mechanisms that provide connectivity between peers supporting different IP versions. In Section 4 we go into transition stages that are important for an ISP. They include launch, backbone, provider access, and complete. In particular, we discuss IPv4-IPv6 transition of the backbone network and provider access networks. Section 5 goes into some technical issues important for an ISP network, including DNS transition, routing, security, and network and service operation. Finally, in Section 6 we give a summary of this document and some conclusions.

2 Motivation

Original motivation for developing IPv6 was the shortage of the address space in IPv4. In [42], authors estimated IPv4 address exhausting by 2004, supposing a very inefficient allocation of IPv4 addresses. Today, this situation does not look so critical. There are

four Regional Internet Address Registries (RIRs), managing a global pool of 91 "/8 blocks". Each of these blocks contains 16 millions IPv4 addresses. From 1999 to 2003, about 20 of them have been consumed. Even with a more rapid growth of IP address consumption, caused by next generation mobile devices and home networking, it will be enough IPv4 address space for next several years. Technologies like Network Address Translators (NATs) help saving IPv4 address demand additionally. However, IPv6 offers by far more address space, allowing each molecule on the surface of the earth to have a unique IPv6 address [39]!

Although the address space was the main motivation, it is not the only benefit of IPv6. Other benefits include:

- more efficient packet routing,
- native support for secure communication,
- improved support for mobile IP,
- serverless autoconfiguration,
- deeper hierarchy and policies for network architecture flexibility, and
- inherent multicast support with increased address space.

Another very important factor pushing demand for IPv6 is the acceptance of IPv6 by international standardization and specification bodies. An example here is the Third Generation Partnership Project (3GPP) that defines the Universal Mobile Telecommunication System (UMTS). The IP Multimedia Subsystem (IMS), which is a part of UMTS Release 5 and above, supports native IPv6 only.

Because of all these reasons, ISPs should start thinking about supporting IPv6 in their networks. However, it is very unlikely that the transition between IPv4 and IPv6 will happen over night. More likely, the transition will be a long process, during which IPv4 and IPv6 will coexist. Figure 1 shows possible transition phases. The main focus of this document are the phases between *IPv4 only* and *IPv4/IPv6 mixed*. For this transition period some mechanisms have been designed to ensure a smooth and stepwise transition. An important issue is the independence between IPv4 and IPv6 connectivity and services. This means that IPv4 services should be available in a network even in the case of IPv6 services failure, and vice versa. The transition between phases *IPv4/IPv6 mixed* and *IPv6 only* will be mainly driven with the high costs of managing both protocols in a network and it is out of scope of this document.

For an ISP, there are several issues that should be considered for the transition. The main motivation for IPv6 from the point of view of an ISP is providing its customers with IPv6 connectivity. However, some ISPs can be motivated with some of the IPv6 benefits such as address space, or lower costs of running an IPv6 network than of running an equivalent IPv4 network [24]. In each case, an ISP has to update its network for IPv6 connectivity, without interrupting IPv4 connectivity and services. Transition mechanisms for an ISP network must in general accomplish the following requirements:

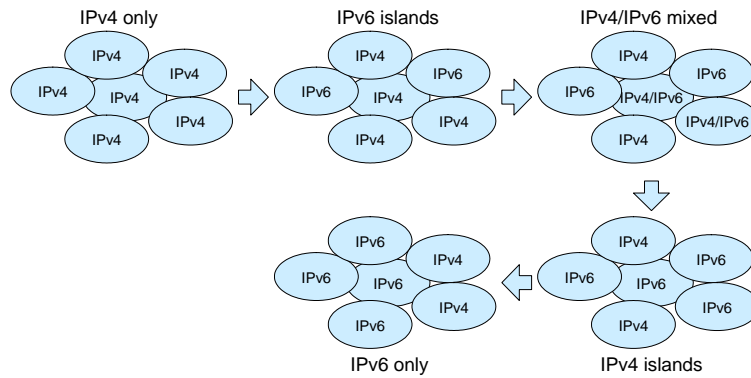


Figure 1: IPv4 to IPv6 transition phases

- They must be simple, as they will be deployed by a large number of ISPs, and
- they must be able to identify a user as an ISP's customer, given that the customers are charged for ISP's services.

An ISP's network is basically composed of a backbone network and several provider access networks. Migration strategies for the backbone are in general different from the strategies for provider access networks. Even strategies for provider access networks are different from each other, depending whether the connected customer network is managed or unmanaged. The best transition strategy also depends on the existing components deployed in the ISP network. For example, if the ISP network already supports MPLS, it can be reasonable to use a migration strategy based on MPLS.

Because of a large set of transition possibilities, it is not easy for an ISP to choose the right transition strategy. This document gives an overview of the most important transition mechanisms that exist today, and some advisement about their suitability for a large ISP network. It should provide a help for ISPs to choose the right strategy for their networks.

3 Existing Transition Mechanisms

As outlined before, the target of the transition considered here is not a total replacement of traditional IPv4, but rather a coexistence of IPv4 and IPv6. During the period of coexistence of these two IP versions, it is possible that IPv4-only nodes, IPv6-only nodes or dual stack hosts communicate. As long as the communicating hosts use the same Internet protocol and all routers in the communication path support this protocol, communication is possible (as it has been over years). If the interacting hosts share the same Internet protocol, but routers in the path do not support it, some kind of tunneling technique has to be applied. If the participating hosts use different IP versions, they require a translation mechanism between the two Internet protocols.

Name	Connectivity	Type	Location
Dual stack	4-to-4 over 4, 6-to-6 over 6	Dual stack	In single ES or ND
SIIT	6-to-4, 4-to-6	Translator	In single ES or ND
Bump-in-Stack (BIS)	4-to-6	Translator	In single ES
Bump-in-API (BIA)	4-to-6	Translator	In single ES
NAT-PT	6-to-4, 4-to-6	Translator	In single ND
MTP	4-to-6, 4-to-6 (multicast)	Translator	In single ND
TRT	6-to-4	Translator	In single ND
SOCKS64	4-to-6, 6-to-4	Translator	Between ES and ND
6over4	6-to-6 over 4	Tunnel	Between ES and ND
ISATAP	6-to-6 over 4	Tunnel	Between ES and ND
DSTM	4-to-4 over 6	Tunnel	Between ES and ND
Configured IP-in-IP	6-to-6 over 4, 4-to-4 over 6	Tunnel	Between ES and ND, two NDs or two ESs
6to4	6-to-6 over 4	Tunnel	Between two NDs

Table 1: Existing IETF transition mechanisms [42]

Network transition is a very complex task, as there exist several applications of each transition mechanism, each with its peculiar pros and cons. A very good overview about IPv6 transition mechanisms can be found in [42]. Table 1 lists the transition mechanisms identified in that article. The location describes whether the device is located in the end-system (ES) or in a network device (ND). We do not describe all of the in Table 1 outlined transition techniques, as not all of them are relevant for ISP networks.

This section is organized as follows: First we describe the operation of two IP stacks, which is called dual stack. Second, we introduce tunneling mechanisms. Subsequently, we give an overview about prevalent translation mechanisms, which offer interworking between different Internet protocols. Migration of domains that use multiprotocol label switching (MPLS) and other translation mechanisms conclude this section.

3.1 Dual Stack

The target of the first step in IPv6 transition is a dual stack core network. Dual stack, which is also known as dual IP layer, provides complete support for both Internet Protocols, IPv4 as well as IPv6 [14]. We can differentiate between dual stack routers and dual stack hosts.

A dual stack router can forward both, IPv4 and IPv6 traffic. Clearly, the router can

forward IPv4 traffic only on links to adjacent IPv4 routers and IPv6 traffic on adjacent IPv6 links. Alternatively, these dual stack routers can additionally act as tunnel entry or exit point, which is used to bridge incompatible networks and described in the preceding subsection.

Dual stack hosts provide IPv4 as well as IPv6 protocol stacks, as illustrated in Figure 2. Therefore, applications can choose "their" protocol stack. IPv4 applications will use the IPv4 protocol stack, and IPv6 applications the IPv6 stack. As socket commands of both IP versions are different, legacy applications that have been designed for IPv4 need to be upgraded to IPv6. Moreover, if an application layer protocol uses IP addresses in the payload – as File Transfer Protocol (FTP), Session Initiation Protocol (SIP) or H.323 do – the corresponding applications need to be updated to be able to handle the longer IPv6 addresses.

IPv4 applications	IPv6 applications
UDP/TCPv4	UDP/TCPv6
IPv4	IPv6
Layer2	
Layer1	

Figure 2: Dual Internet protocol stack

Note that a full dual stack core network solely enables the communication between two IPv4 hosts, as well as between two IPv6 hosts – it does not provide a solution to IPv4 hosts that need to communicate with IPv6 hosts and vice versa.

As mentioned earlier, dual-stack is the target of core network migration, as all routers are capable to process IPv4 and IPv6 in parallel. Nevertheless, dual stack is expensive as all routers have to upgrade. It is also possible to use this event to buy totally new routers that are newer and therefore offer a higher performance. Anyhow, these investments for IPv6-capable equipment are not justified for only few IPv6 users. At the other hand, when the amount of time that has to be spent on the configuration of IP tunnels traverses a certain threshold, operators should think about migration to a dual stack core network.

Summarized, a dual stack network has high capital expenditures (CAPEX) and operational expenditures (OPEX) at the installation phase. The future will show whether OPEX costs of dual stack operation are nearly equal to IPv4 operation or significantly higher. Anyhow, the operator has to afford the maintenance (which involves management and security) of two IP versions.

3.2 Tunneling

In general, tunneling allows to bridge incompatible networks. It denotes the transport of IPv6 packets encapsulated in IPv4 packets (or vice versa) as illustrated in Figure 3. The tunnel entry point encapsulates IPv6 packets into IPv4 packets. The tunnel exit point

decapsulates the IPv6 packets and forwards them towards the target. Even when tunnel entry and exit point in Figure 3 are intermediate devices, it is possible that sending or receiving hosts act as tunnel entry or exit point.

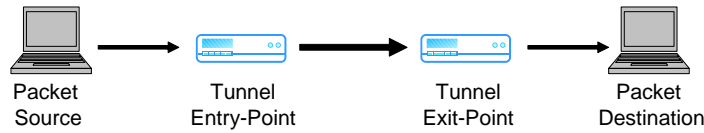


Figure 3: Communication tunnel

Figure 4 illustrates the encapsulation process that has to be performed at the tunnel entry point. If the original packets are IPv6 packets, the tunnel entry point prepends e.g. the IPv4 header and forwards the packets towards the tunnel exit point. The resulting packets consist therefore of the original IPv6 packets and the prepended IPv4 headers. If the original packets are IPv4 packets that have to be tunneled over IPv6, the tunnel entry point prepends the IPv6 header and optional IPv6 extension headers before it forwards the packets. Decapsulation is the reverse process, the tunnel exit point discards the outer header and forwards the encapsulated packet.

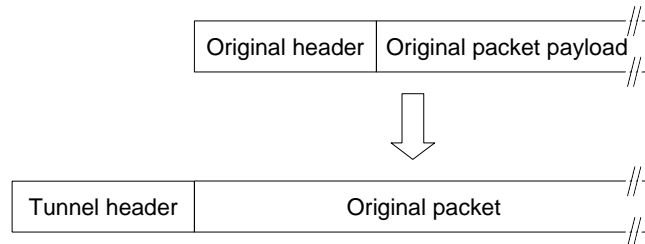


Figure 4: Encapsulation of tunnel packets

Through tunneling mechanisms it is possible that potential attackers circumvent ingress filtering [13]. Therefore, decapsulating routers should only forward packets encapsulated by well-known, trustworthy IPv4 source addresses.

There are several interesting issues that should be noticed when using tunnels in network migration. First, the encapsulated protocol is not aware of the tunnel hop count. From the encapsulated protocol's point of view this means that a tunnel consisting of several routers looks like just one hop. Second, it is possible to use recursive encapsulation, i.e. a tunnel inside a tunnel. In this scenario, the tunnel entry point of the inner tunnel encapsulates packets that have already been encapsulated by the tunnel entry point of the outer tunnel. The tunnel exit point of the inner tunnel decapsulates the packets and forwards them towards the outer tunnel exit point, which once more

decapsulates the packets to their original format. An example of recursive tunnels are IP security tunnel packets once again encapsulated in packets of another IP version. The third issue regarding tunneling concerns path maximum transmission unit (MTU). In IPv6, end hosts, and not intermediate devices, handle path MTU discovery and fragmentation. Finally, if an error occurs inside the tunnel, ICMP error messages need to be converted between the two Internet protocols.

The advantage of using IPv6 tunnels over an existing IPv4 infrastructure is the already well-tuned IPv4 infrastructure, which should be able to carry the additional IPv6 traffic. Potential drawbacks are that denial-of-service (DoS) attacks or other implications of the IPv4 production network influence also IPv6 traffic. However, as IPv4 works quite well for the moment, this stability should be sufficient for IPv6 too. Indeed, tunneling in core networks may not be the ideal solution. As IPv6 traffic is expected to grow, native IPv6 transport would be a permanent solution.

The rest of this section describes the most commonly used tunneling techniques. We start with an investigation of configured IP tunnels. Subsequently, we describe two automatic tunneling techniques, namely tunnel broker, 6to4 and Teredo.

3.2.1 Configured IP tunnels

Statically configured IP tunnels give the provider the highest control over the tunneled IP traffic. Each tunnel has to be configured at the tunnel entry and exit point manually. Especially for a high number of IPv6 customers, this consumes much time and effort, which results in high OPEX costs. To support very early IPv6 customers, configured tunnels are a sound solution, but as the demand grows, a migration to an automatic tunneling transition technique (or directly to dual stack) should be considered.

3.2.2 Tunnel Broker

The tunnel broker allows users with a dual stack host to connect to the provider's IPv6 network via a dynamic tunnel [12]. The user residing on a dual stack workstation contacts a web server. Therefore, it provides information that is necessary to configure the tunnel. Subsequent, the user downloads an operation system (OS) specific script that establishes the IP tunnel to the IPv6 network of the provider.

Figure 5 illustrates the establishment of a tunnel using a tunnel broker. First, the user downloads the tunnel configuration script for his dual stack host from the tunnel broker web server. Second, this server informs the tunnel broker tunnel server about the new host. Third (and finally), the host establishes a dynamic tunnel to the tunnel broker tunnel server. Operating a tunnel broker, it is desirable to check periodically the status of the tunnel clients. If there exist hosts that are not reachable, the tunnel broker can free the occupied resources.

There exist three popular tunnel brokers. Hexago¹ that allows dual-stack home users

¹<http://www.hexago.com>

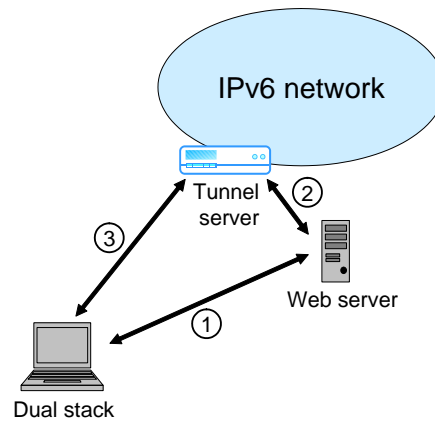


Figure 5: Tunnel broker

to connect to the IPv6 backbone, SixXS², which operate tunnel broker Points of Presence (POPs) for European ISPs free of charge and the open source OpenVPN IPv6 tunnel broker. The German JOIN project provides detailed information about how to set up such an OpenVPN IPv6 tunnel broker³.

To be sure that only authorized parties use the tunnel broker, it is essential to authorize tunnel broker users. Therefore, each operator that provides a tunnel broker should maintain a user database (DB). As each user downloads the script for the tunnel configuration, the provider does not need to configure each tunnel manually. At the other hand, each user that needs an IPv6 tunnel has to be registered in a DB. As DB administration is much simpler than tunnel configurations, the operation of a tunnel broker saves OPEX costs compared to manually configured tunnels when the number of IPv6 customers is large.

3.2.3 6to4

6to4 is a mechanism that can be used to connect IPv6 islands over the IPv4 Internet [5]. This means that IPv6-only hosts can use this mechanism to communicate with other IPv6-only hosts. The automatic tunneling mechanism of the 6to4 transition mechanism bases on the embedded IPv4 address of the 6to4 tunneling router in the IPv6 addresses of the IPv6 network. Note that because of this circumstance the IPv4 address should be static long-lived, as an IPv4 address change requires all IPv6 addresses in the 6to4 site to change accordingly.

Figure 6 shows the 6to4 address format. The prefix for the 6to4 address is 2002::/16. The following 32 bits contain the IPv4 address of the 6to4router. The remaining bits are used for the subnet and interface ID of the respective IPv6 end host. To connect a new IPv6 island to other islands, it is not necessary to change any of the existing configurations, as all IPv4 packets carrying IPv6 traffic will always automatically find their

²<http://www.sixxs.net>

³http://www.join.uni-muenster.de/Dokumente/Howtos/Howto_OpenVPN_Tunnelbroker.php

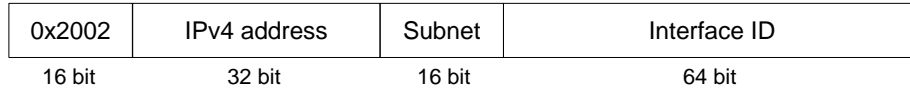


Figure 6: 6to4 address format

destination because of the embedded IPv4 address. The concept of the 6to4 tunneling mechanism is illustrated in Figure 7.

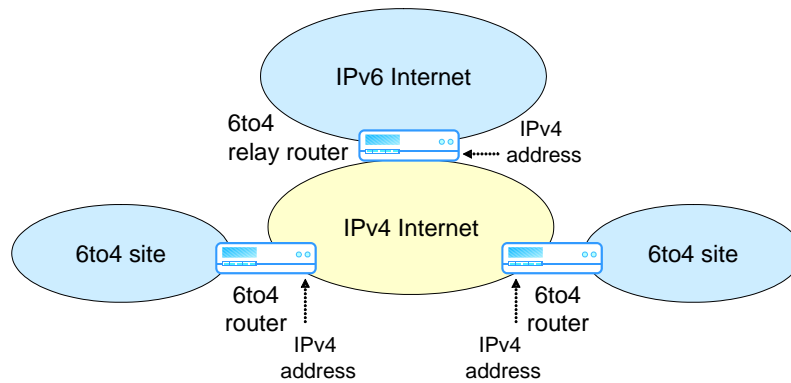


Figure 7: 6to4 tunneling mechanism

Even if this transition technique is well suited for IPv6 islands – 6to4 sites can easily communicate with other 6to4 sites, communication with native IPv6 (i.e. not 6to4 sites) is not so easy, as these IPv6 islands can only accept IPv6 traffic with the prefix `2002::/16`. Therefore, the 6to4 sites need a 6to4 relay router that connects the native IPv6 global network to the IPv4 global network. All 6to4 routers send the native IPv6 traffic (whose destination address does not have the `2002::/16` prefix) to the 6to4 relay router that forwards the traffic towards the global IPv6 network. Contrariwise, the 6to4 relay router encapsulates all traffic destined to a 6to4 site into IPv4 and forwards them to the appropriate 6to4 router. The reachability of a 6to4 relay router can be ensured by configuration on the 6to4 routers or by using the `192.88.99.0/24` IPv4 anycast address [16] for the 6to4 relay router .

There are several security issues concerning 6to4. The most obvious is that 6to4 routers have to accept the traffic from all 6to4 relays to guarantee connectivity. At the other hand, bogus 6to4 relays may inject traffic and generate a Distributed Denial of Service (DDoS) attack. Other security issues regarding 6to4 and 6to4 relays are discussed in [1].

Application of 6to4 is a quick solution to provide customers with IPv6 connectivity. Admittedly it is necessary to provide (or use an external) 6to4 relay router to be capable of communicating with native IPv6 networks. The application of 6to4 also requires a detailed analysis of all known security threats. Moreover, 6to4 cannot be used with

private IPv4 addresses. Therefore, in an environment that relies on a Network Address Translator (NAT) [RFC1631], the NAT device needs to be collocated with the 6to4 router.

3.2.4 Teredo

Teredo [17] is an automatic tunneling technique that provides unicast IPv6 connectivity for dual stack hosts that may reside behind one or even multiple IPv4 Network Address Translators (NATs). It is a similar technique to 6to4 (see Section 3.2.3), but it additionally supports private addresses that can be used together with NATs.

To enable communication through a NAT, Teredo hosts tunnel IPv6 packets encapsulated in IPv4 UDP packets through the NAT device towards a Teredo Relay. Most NAT devices allow outgoing UDP traffic. Therefore, they do not inspect the content of UDP packets, which is a proper IPv6 packet. To be able to receive messages, it is essential to keep the NAT binding alive. This is achieved through Teredo bubble packets. Figure 8 illustrates the structure of a Teredo packet. Teredo bubble packets do not carry IPv6 payload, they consist solely of IPv4, UDP and IPv6 header.

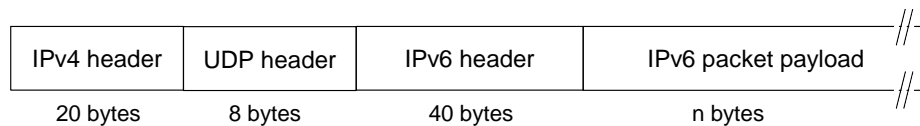


Figure 8: Teredo packet format

Teredo is based on three main components: Teredo client, Teredo server and Teredo relay. As described previously, a Teredo client needs to support both, IPv4 and IPv6. It receives an IPv6 prefix from a Teredo server and acts as tunnel entry and exit point. The Teredo server assigns IPv6 addresses to its Teredo clients. It listens on UDP port 3544 for incoming traffic. Teredo relays are IPv4/IPv6 border routers that relay IPv4 UDP traffic between Teredo clients and native IPv6 traffic of IPv6-only hosts residing in the IPv6 Internet. Therefore, they act as tunnel end points for IPv6 packets tunneled over IPv4 UDP. An interesting point is that Teredo relays may also provide interworking with other transition mechanisms as 6to4 (see Section 3.2.3). A special component is a Teredo host-specific relay. This is a dual stack host connected to both, IPv4 and IPv6 Internet. Because of its IPv4 connection, the host-specific relay does not need a Teredo relay. It can send proper encapsulated IPv6 packets directly over the IPv4 Internet towards Teredo clients. Figure 9 demonstrates the Teredo architecture.

To allow native IPv6 hosts to connect to Teredo hosts, it is essential that the IPv6 network is capable of routing Teredo packets towards the nearest Teredo relay. Therefore, the IPv6 addresses of Teredo clients need a certain format that is shown in Figure 10. Teredo routing relies on the fact that the Teredo server assigns its Teredo clients IPv6 addresses. These addresses contain the IP address and the port of the NAT device, so that IPv4 routing delivers the encapsulated IPv6 packets to the proper destination. In

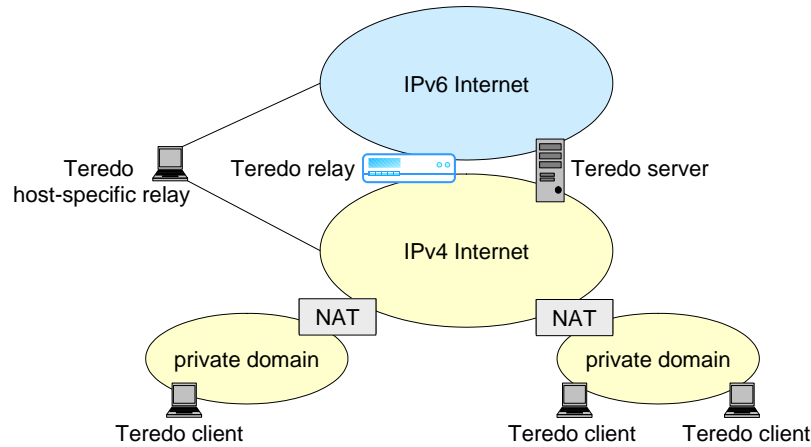


Figure 9: Teredo architecture

fact, the Teredo relay generates UDP and IPv4 headers on receiving traffic from native IPv6 hosts. The IPv4 destination address – which means the appropriate NAT device – can be extracted from the Teredo address (Figure 10). A good overview about Teredo can be found in [10].

Teredo prefix	Teredo server IPv4 address	Flags	NAT port	NAT IPv4 address
32 bit	32 bit	16 bit	16 bit	32 bit

Figure 10: Teredo address format

Similar to 6to4, there are many security concerns about Teredo. Security risks include spoofing, man-in-the-middle attacks, and denial of service attacks. Detailed security considerations can be found in [17]. Note that because of the numerous kinds of different NATs, Teredo is a very complex transition technique.

3.3 Translation

As mentioned earlier, dual stack and tunneling provide the infrastructure for communication between hosts with the same IP version. To support the communication between hosts using different IPs, IP translation is necessary. This translation can reside on different levels. Translation can be performed at application level using an Application Level Gateway (ALG), at protocol level using Network Address Translation – Protocol Translation (NAT-PT), at transport level using a Transport Relay Translator (TRT), or at the socket layer using SOCKS64. We describe all the enumerated translation techniques subsequently. Most of the mentioned translation techniques are based on the Stateless IP/ICMP Translation (SIIT).

3.3.1 Stateless IP/ICMP Translation

SIIT [34] is a bidirectional translation of IP and ICMP packets between versions 4 and 6. For IP packets, the translation covers only the header, for ICMP the whole packet is translated. It is clear, that most of the optional IPv6 extension headers cannot be translated, the only exception is the fragment header. SIIT is the basis of higher layer transition in NAT-PT.

3.3.2 Application Level Gateway

An ALG is an intermediary device located between two communication partners. It acts mostly as an application proxy between IPv6-only clients and legacy servers that offer their services only via IPv4. This means that application-specific information is substituted in the application layer at the ALG. The most common variant is a Domain Name Service (DNS) ALG that is necessary for NAT-PT.

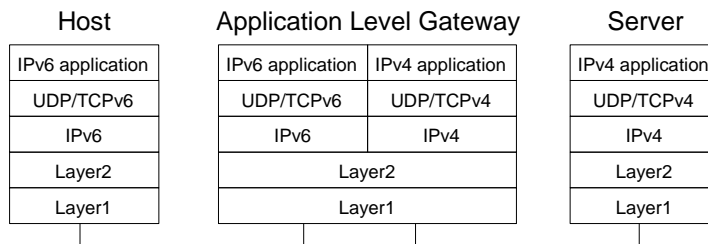


Figure 11: Application level gateway

Figure 11 illustrates the architecture of an ALG. The host in the example in Figure 11 uses IPv6 and the server IPv4, although this could be contrary as well. The host sends an IPv6 request towards the ALG. There, the ALG removes all IP/TCP/UDP headers, translates the payload from the IPv6 application format to an IPv4 application format and forwards the request using IPv4. The server sends the response to the ALG, which translates the payload of the response and sends the response back to the client using IPv6.

There are several security issues regarding an ALG. First, the ALG breaks the end-to-end security scheme, as it is a man-in-the-middle (MITM). Second, if the server has no application level authentication, it assumes that the ALG is the client, whereas the real client is hidden behind the ALG.

Note that an ALG is application-specific, i.e. it is necessary to deploy an ALG for each application (or one ALG that translates all deployed applications). An ALG is most likely a single point of failure – if it fails, none of the clients can reach a server that does not support its IP version.

3.3.3 Network Address Translation - Protocol Translation

Based on SIIT (see Section 3.3.1), NAT-PT [41] is a stateful IP version translator. It uses a pool of IPv4 addresses for its IPv6 client hosts. Using a DNS ALG, NAT-PT supports communication initiated by IPv4 nodes, as well as IPv6 nodes.



Figure 12: IPv4 compatible IPv6 address

If an IPv6 node initiates the communication, the sender encapsulates the IPv4 destination address into an IPv6 address - therefore the destination address is an IPv4 compatible IPv6 address as illustrated in Figure 12. Communication initiated by an IPv4 host is triggered by the DNS ALG that is co-located with NAT-PT. The result of this DNS query is a temporary IPv4 address that is assigned to the IPv6 host for the duration of the communication. In both directions, the NAT-PT serves as communication proxy that uses the SIIT mechanism to translate the passing packets. Figure 13 shows the architecture of NAT-PT. There exist variants of NAT-PT that also translate the transport identifiers (TCP and UDP port numbers, as well as ICMP identifiers). This kind of NAT-PT is called Network Address Port Translation - Protocol Translation (NAPT-PT).

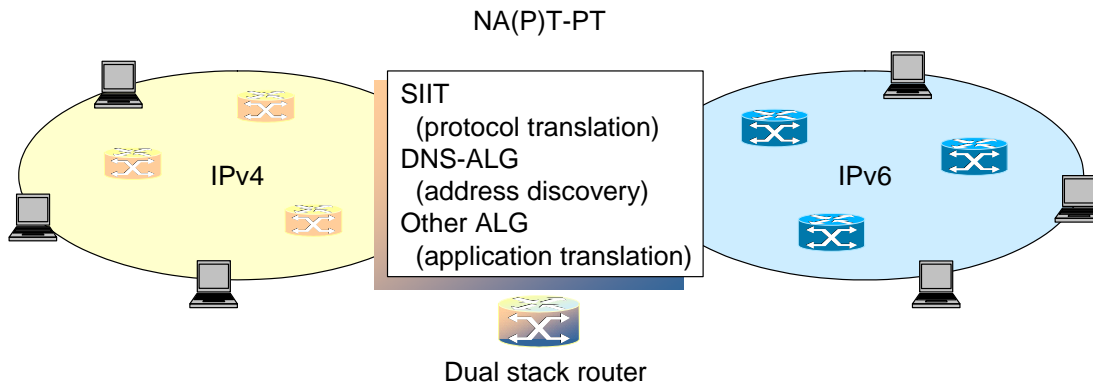


Figure 13: Network Address (Port) Translation - Protocol Translation

Note that in a NAT-PT cluster, each session must traverse the same NAT-PT device because of the stateful behavior of the translation mechanism. For each application where the payload between IPv4 and IPv6 is different, NAT-PT requires a collaborating ALG.

3.3.4 Transport Relay Translator

Another possibility to traverse between different IP versions is the application of a TRT [21]. An IPv6 host may initiate a request using an IPv4 compatible IPv6 address illustrated in Figure 12 for the communication target address. The request will be routed through the TRT that resides on the IPv4/IPv6 domain border. There, the TRT terminates the IPv4 UDP or TCP session and initiates a new IPv6 UDP respective TCP session. Figure 14 illustrates the architecture of a TRT.

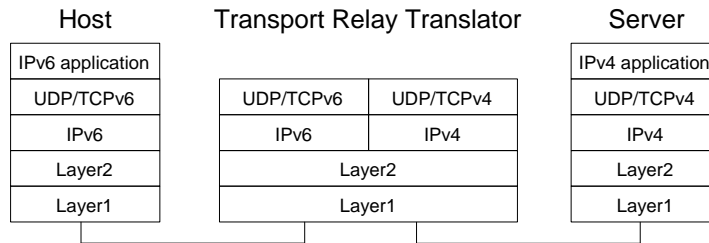


Figure 14: Transport Relay Translator

As the TRT acts as MITM, it breaks end-to-end security. Additionally, it hides the real source address of the client. Therefore, it also breaks source address based authentication (which is anyway a bad solution for authentication).

As the application level is transparent for the TRT, applications using layer three IP addresses as e.g. SIP need an additional ALG (see Section 3.3.2) to work across a TRT.

3.3.5 SOCKS64

For enterprises that already use SOCKS [25], application of SOCKS64 [23] is the preferred way to translate between different IP versions. Figure 15 illustrates the SOCKS architecture. The socks library intercepts all socket calls and DNS queries and forwards them to the SOCKS gateway. As the SOCKS gateway has full control over all network based communication, it can easily differentiate between communication that needs to be translated and communication where both, sender and receiver, share the same IP.

To update a legacy SOCKS infrastructure, it is necessary to update the SOCKS library as well as the SOCKS gateway with newer versions that also perform translation between IPv4 and IPv6.

Note that SOCKS64 provides security between the client and the gateway, as well as between the gateway and the destination. Admittedly, it does not provide end-to-end security between the client and the destination, i.e. it breaks IPSec.

3.4 MPLS

If the core network already operates using Multiprotocol Label Switching (MPLS) [38, 37], it may be desirable to keep the MPLS infrastructure also for IPv6 operation. Gen-

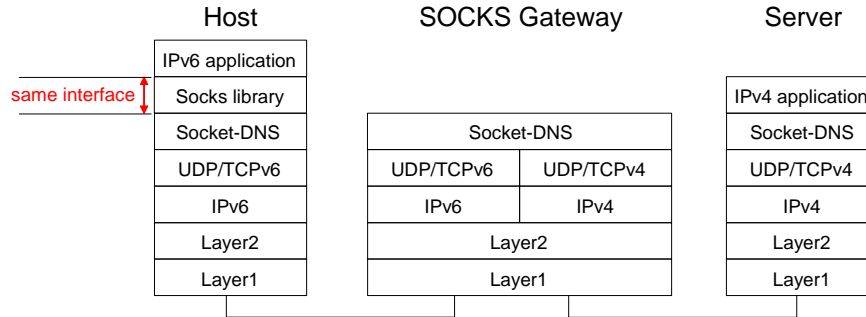


Figure 15: Socket based IP gateway

erally, there exist several possibilities to enable IPv6 over MPLS. We already mentioned various tunneling mechanisms that are transparent for MPLS. Additionally, there exist transition mechanisms that slightly modify an existing MPLS infrastructure. First we describe the most desirable approach, native IPv6 forwarding in each Label Switching Router (LSR). The second approach uses a special case of BGP tunneling to route IPv6 packets over an IPv4 MPLS core network. Finally, operators can use layer 2 tunneling over their existing MPLS infrastructure. We conclude this subsection with a brief description of layer 2 tunnels over MPLS.

3.4.1 Native IPv6 over MPLS

Native IPv6 over MPLS means that all MPLS routers use IPv6 as network protocol and MPLS for routing decisions. To enable native IPv6 over MPLS, an operator would need to update the existing Label Switching Routers (LSR) with IPv6 support. This support needs to be twofold: First, the routers need IPv6 capable network interface cards supporting the IPv6 protocol stack, and second MPLS with its label distribution needs to support IPv6 addresses. Unfortunately only a minority of router vendors support native IPv6 over MPLS. According to the 6net transition cookbook for ISP and backbone networks [3], just ZebOS and AYAME support native IPv6 over MPLS.

3.4.2 IPv6 over IPv4 MPLS Core

Using MPLS, it is possible to adapt just the edge routers with the new IP. Using a special case of BGP tunneling [35], LSR can distribute IPv6 prefixes using BGPv4 over IPv4. The old, core routers remain nearly unchanged in the MPLS core network. Cisco called this functionality IPv6 Provider Edge Router (6PE), which is now the common notation.

6PE assumes that the MPLS network consists of so-called provider (P) routers that switch MPLS packets and do not need to investigate IP headers. Additionally, at the edge of the MPLS network there are provider edge (PE) routers that inspect the IP header, mark the packet with an MPLS label and route it towards a P router. Customers

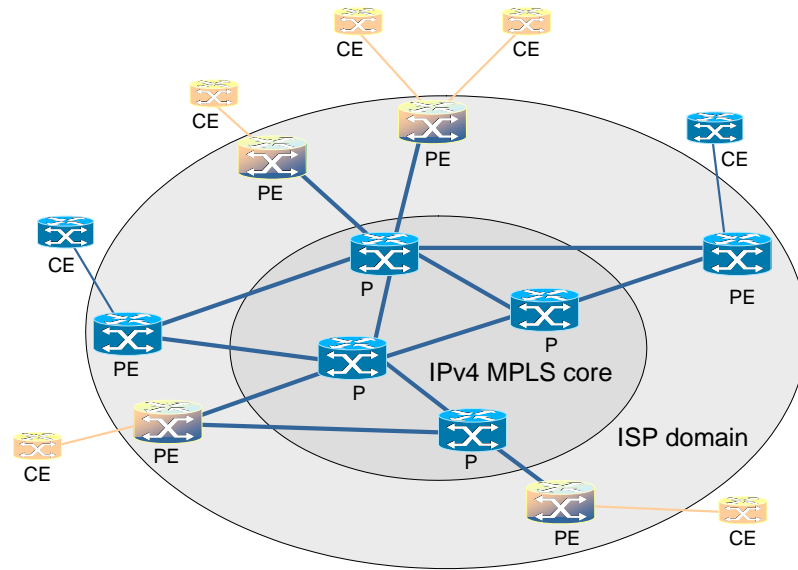


Figure 16: IPv6 Provider Edge Router (6PE) architecture

are connected via customer edge (CE) routers, that are connected to PE routers. Figure 16 illustrates the architecture of 6PE. Light-colored links illustrate IPv6 links, dark links denote IPv4 links. Similarly, IPv4 routers are dark (note that all P routers are IPv4 routers), IPv6 routers are light-colored and dual stack routers are mixed.

6PE has the advantage, that only those PE routers, that are connected to IPv6 CE routers have to be upgraded to IPv6, whereas all P routers (and PE routers without connected IPv6 CE routers) may still operate using IPv4.

3.4.3 Layer 2 tunneling over MPLS

Using MPLS, it is possible to transport a variety of protocols. In most cases, MPLS will be used to transport IP packets. Nevertheless, it is possible to transport layer 2 frames over MPLS [33, 32]. In principle, MPLS may be used to carry Frame Relay [22], ATM AAL5 [30] and Ethernet Protocol Data Units (PDUs) [31], as well as Synchronous Optical Network (SONET)/ Synchronous Digital Hierarchy (SDH) circuit emulation [27].

Figure 17 illustrates the protocol stack of layer 2 tunneling over MPLS. The Provider Edge (PE) router needs to encapsulate the layer 2 frames into MPLS and send it through the MPLS backbone. The egress PE router of the MPLS domain removes the MPLS label and forwards the encapsulated layer 2 frame on top of layer 1 towards the destination.

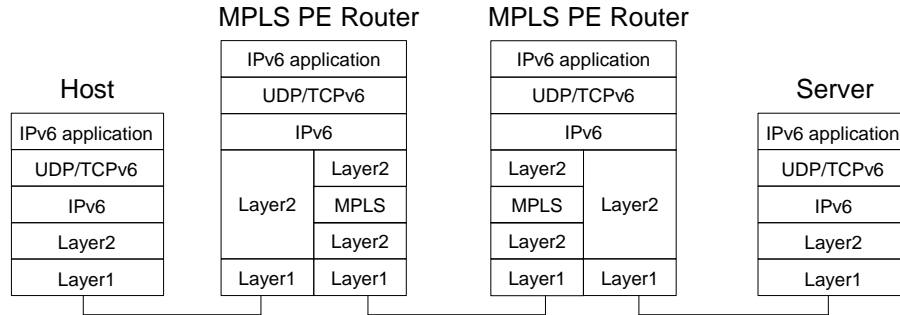


Figure 17: Protocol stack of layer 2 tunneling over MPLS

3.5 Other Translation Mechanisms

There exist two other translation mechanisms that might be interesting for operators under certain circumstances. We describe here IPv6 transport over an Asynchronous Transfer Mode (ATM) network. Finally, it is possible to build a new, separated infrastructure solely for IPv6.

3.5.1 ATM

IPv6 over ATM can be seen as a solution, if an operator already operates an ATM network. Even if (according to [3]) there is no National Research and Education Network (NREN) IPv6 production network using an implementation of IPv6 over ATM, it would be possible to run IPv6 over parallel Permanent Virtual Circuits (PVCs) or soft PVCs. ATM looks like a good solution with relatively low costs for an operator.

3.5.2 Separate IPv6 Network

A totally different approach is to build a separated, new IPv6 network beside the existing IPv4 infrastructure. This is an expensive approach, as the provider needs a new network.

The advantage of this approach is that IPv6 traffic will not interfere with IPv4 traffic. The disadvantage is the cost of the additional equipment, the maintenance costs and the costs for the customers, which have to connect (and afford) an additional line to the IPv6 core.

4 Transition Stages

This section discusses possible transition scenarios for an ISP provider. It is based on the work of the IETF IPv6 Operations (v6ops) working group⁴. They have identified four scenario areas:

⁴ IPv6 Operations (v6ops) Working Group of the Internet Engineering Task Force, <http://www.ietf.org/html.charters/v6ops-charter.html>

- ISP networks,
- enterprise networks,
- unmanaged networks, and
- cellular networks.

The goal is to provide guidelines to network operators and users for IPv6 deployment within existing IPv4 networks. As mentioned in Section 2, in this deliverable we will concentrate on ISP networks. The current v6ops proposal for this environment [26] is still in draft state, this means the work is in progress. This document gives a study of scenarios and an analysis for introducing IPv6 into large ISP networks.

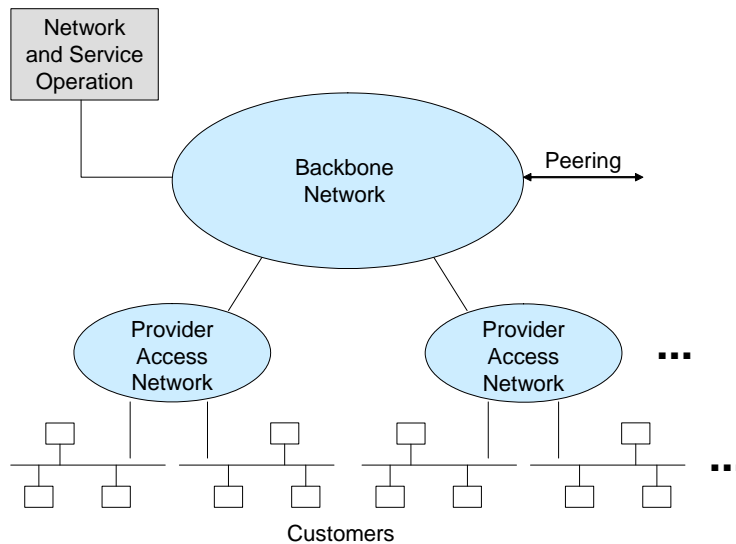


Figure 18: General view of an ISP network topology

Figure 18 shows a general view of an ISP network topology. We can notice one backbone network and provider access networks⁵, each of which connects one or more customers. The backbone network provides connectivity between provider access networks and to other ISPs' networks. Entities residing in the backbone network are core routers, border routers that exchange routing information with other ISPs, and a part of the Provider Edge equipment (PE). The other part of PE is placed in the provider access network. It also includes Customer Premises Equipment (CPE) and the last hop link, providing connectivity to the customers. Finally, the *Network and service operation* part is responsible for services, like management, billing, and accounting, in this ISP network.

⁵ The Internet Draft [26] uses the term *customer connection network* for the network that connects customers with the backbone network. In this document, we will always use the Cisco terminology, according to which all the networks that belong to a provider starts with the word provider, and all the networks that belong to customers with the word customer. Therefore, we call this network *provider access network*.

4.1 Definition of Stages

Depending on the network segment where the IPv4-IPv6 transition occurs, several scenarios are possible. The mentioned v6ops proposal [26] identifies the following possible transition stages:

- Stage 1: *Launch* – It represents an ISP with an IPv4 network and IPv4 customers. This is the starting point for most of the ISPs. To provide IPv6 connectivity to its customers, an ISP can move from this stage to any other stage.
- Stage 2a: *Backbone* – In this stage, there is IPv6 support additionally to IPv4 support in the backbone network, whereas provider access networks support IPv4 only. Customers are provided with IPv6 connectivity using a tunneling mechanism through the provider access networks.
- Stage 2b: *Provider access*⁶ – Here, provider access networks support both, IPv4 and IPv6, and the Backbone network supports IPv4 only. IPv6 connectivity through the backbone network is provided by tunneling.
- Stage 3: *Complete* – This stage is the final step for the IPv6 deployment. The backbone network, as well as the provider access networks, support IPv4 and IPv6.

The main difference between Stages 2a and 2b is that in Stage 2b customers do not need to support both IPv4 and IPv6, but only IPv6.

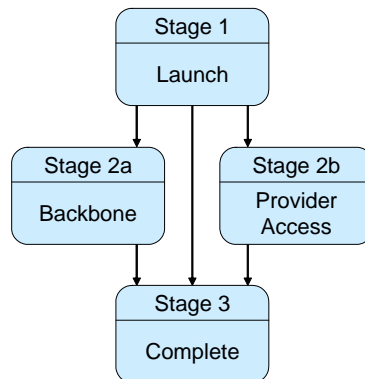


Figure 19: Possible transitions between stages

As Figure 19 shows, there are several ways for an ISP to go through these stages. It is also possible to combine them, for example to have an IPv6 capable backbone network and some of the provider access networks that support IPv6. This scenario would be a combination of Stages 2a and 3.

Note that each stage after Stage 1 provides both, IPv4 and IPv6 connectivity to ISP's customers. In the following subsection, we will discuss the transition possibilities for the backbone network and for the provider access networks.

⁶In the Internet Draft this stage is called *customer connection*.

4.2 Backbone Transition

The starting point for the backbone transition is a backbone that supports IPv4 only, and the final point is a dual-stack backbone. There can be several steps between these two points, since it is very unlikely that the transition occurs from one day to another. On the other side, it is also possible to jump directly into the final step (dual-stack backbone), replacing (or upgrading) all the routers in the backbone with dual-stack routers.

The choice of the best strategy for the backbone transition depends on the number of customers requiring IPv6 connectivity. At the beginning, we expect just a few such customers. In that case, statically configured IPv6 over IPv4 tunnels, described in Section 3.2.1, are a cost-effective solution. The existing, well-tuned, IPv4 infrastructure is used for transport of IPv6 packets, and the existing IPv4 services are not interrupted. However, since the tunnels are configured manually, it is possible that IPv6 packets do not take the optimal route between tunnel end-points. This is because one IPv6 hop usually spans several IPv4 hops.

As the number of IPv6 customers grows, the configuration of static tunnel will become too time-consuming and therefore inefficient. At this point, the ISP can start using automatic tunnels. The 6to4 mechanism, discussed in Section 3.2.3, is a possible solution for the backbone. On the other side, solutions based on a tunnel broker (Section 3.2.2), do not suit for the backbone network, since they require end users' interactions for the tunnel establishment.

After the introduction of automatic tunnels, some router in the backbone network can be replaced (or upgraded) with dual-stack routers. This process should continue until all the core routers are dual-stack capable. The need for automatic tunnels will slowly disappear, since always more and more provider access networks will be connected through dual-stack routers.

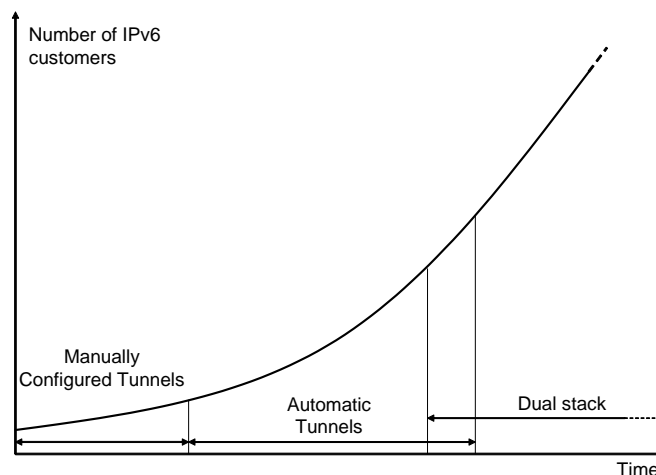


Figure 20: Backbone transition mechanisms depending on the number of IPv6 customers

Figure 20 illustrates the backbone transition process. It is based on an estimation of

the number of IPv6 customers. In fact, the most important factor for the time scale is the number of customers that require IPv6 connectivity. Therefore, an ISP can stay for a long time in the first step (manually configured tunnels), if the number of IPv6 customers remains small.

A special case for the transition of the backbone networks represents ISPs that already have deployed MPLS in the backbone. For these ISPs, a cost-effective solution may be IPv6 over MPLS in the backbone. In Section 3.4.2 we have discussed the 6PE mechanism that is suitable for the backbone network. In this way, only the edge network routers need to be IPv6 aware, whereas the core routers do not need to be changed. The overhead of this tunneling technique is significantly less than the overhead of IPv6 over IPv4 tunneling (8 Bytes vs. at least 20 Bytes). Given that this solution scales very well, it can exist for a long time, even when the number of IPv6 customers becomes large. As mentioned before, this is only suitable for ISPs that already have deployed MPLS in the backbone network. We do not recommend, however, to deploy MPLS, with all its complexity, purely to introduce IPv6.

4.3 Provider Access Transition

A provider access network comprises in general a few PEs that provide connectivity to many CPEs. It can be based on several technologies, depending on the size and required bandwidth of the connected customer networks. These technologies include, but they are not limited to dial-up, cable, and Digital Subscriber Line (DSL). In this document, we use the term *customer network* for end sites that belong to customers, and *provider access network* for the network that connects customer networks with the backbone. This is depicted in Figure 21.

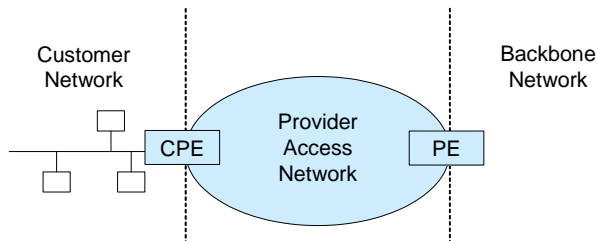


Figure 21: Provider access network and customer network

Similar to the backbone transition (Section 4.2), provider access network transition can include several steps. Also here, these steps are optional, and an ISP can go directly into the last step – dual stack provider access networks. The first transition step is providing IPv6 connection using one of the tunneling mechanisms. Which tunneling mechanism can be used depends on whether the customer network is managed or unmanaged. Usually, small customer networks are unmanaged, whereas large customer networks are managed. We will go into differences between these two kinds of networks later in this section.

An important issue for provider access networks is the presence of NATs between two tunnel endpoints. This should be avoided, since several tunneling techniques are not NAT compatible. In some cases, the CPE acts as a NAT. Terminating tunnels at the CPE, avoids the NAT transversal problem in such cases. On the other hand, the CPE has to support the applied tunneling mechanism. In the following, we discuss possible tunneling mechanisms for provider access networks depending whether the customer network is managed or not.

4.3.1 Unmanaged Customer Networks

Transition scenarios for unmanaged networks are discussed in [19]. This document describes an unmanaged customer network composed of a home gateway and several hosts. It investigates the following cases:

1. a gateway with no IPv6 support,
2. a dual-stack gateway connected to a dual-stack ISP,
3. a dual-stack gateway connected to an IPv4 ISP, and
4. a gateway connected to an IPv6 ISP.

Case 4 is not relevant for us, given that the goal of the transition is not a native IPv6 ISP, but rather a dual-stack ISP. In Case 2, both gateway and ISP network are dual-stack enabled, allowing hosts to be IPv4, dual-stack, or even IPv6 only.

For Case 1 and Case 3, a tunneling mechanism is necessary. In [18], different tunneling techniques for unmanaged networks are discussed. In general, tunneling solution can be divided into *automatic* and *configured*. Automatic solutions include 6to4 (see Section 3.2.3) or Teredo (see Section 3.2.4). They work with a minimal support of IPv6 in the ISP network. Configured solution relies on configured tunnels with explicitly defined tunnel routers. They may work with a tunnel broker (see Section 3.2.2) or not. However, when configured tunneling is not possible, for example because of dynamic IPv4 addressing, some automatic mechanism is to be used. Options are a layer 2 tunneling protocol (L2TP), or a tunnel configuration service like [2].

4.3.2 Managed Customer Networks

Managed customer networks are in general simpler for transition than unmanaged networks. They are normally large and they have static IPv4 addresses. A good transition solution for such networks are configured tunnels (Section 3.2.1). It is also possible to use a tunnel broker (Section 3.2.2), but this is typically unnecessary, given that configured tunnels are simpler. Using automatic mechanisms mentioned in the last section, 6to4 and Teredo, is not recommended here.

4.3.3 General Issues for Provider Access Transition

From the point of view of an ISP, there are several important issues regarding the transition of provider access networks. Here we want to address three of them, namely access control, traceability, and ingress filtering.

An ISP usually does not provide network connectivity to its customers for free, but rather customers are charged for this service. Consequently, there must be a mechanism for customer authorization and access control. Given that this is not a new issue caused by the introduction of IPv6, it already exists sophisticated authorization and access control mechanisms for IPv4 connectivity. The most simply way for the transition is to automatically provide access to IPv6 services to the customers of IPv4 services. Otherwise, the ISP has to perform an IPv6 access control in parallel to the IPv4 access control. Note that authorization and access control traffic can use both, IPv4 and IPv6 connectivity. For example, IPv6 access control traffic can be transmitted over IPv4 and vice versa.

Traceability is another important feature for an ISP. Usually, ISPs are able to trace the origin of the customer traffic coming into their networks, and this must also be ensured for the IPv6 traffic. The ISP has to store bindings between customer and IPv6 addresses (or address ranges), even for the customers connected with tunnels. A possibility is to map DHCP responses to the physical connection of a customer, and to store these bindings in a database. Another possibility is to provide the customers with static addresses.

Ingress filtering has several purposes, including traceability and security, preventing distributed Denial of Service (DoS) attacks, preventing access with spoofed addresses, and preventing customer access to the management system. Ingress filtering is already deployed by most of the ISPs for IPv4 services, and it must also be available for IPv6 connectivity. However, if the traceability for IPv6 customers is provided, ingress filtering can be based on the same mechanisms as for IPv4 customers [13].

4.4 Examples

In this section we outline some examples of ISPs that already offer IPv6 connectivity.

In Austria, ATnet offers native IPv6 access⁷ to Vienna Internet eXchange (VIX). ATnet participates on 6Net project and supports IPv6 connectivity without tunneling. Additionally, they offer "IPv6 tunnels for everyone". This means that any user, event those without an ATnet dedicated line, can obtain IPv6 connectivity over a tunnel to ATnet. They have also deployed IPv6 capable DNS servers and they also offers IPv6 connectivity to ISPs by allocation of whole RIPE IPv6 networks, such as /48 networks.

In Germany, the first ISP that offered IPv6 connectivity was Spacenet⁸. They offer native IPv6 over ADSL access technology. IPv6 customers are provided with static IPv6 address, and each customer, independent on the size, is provided with a /48 IPv6

⁷<http://www.atnet.at/produkte/internetzugang/ipv6.html>

⁸<http://www.space.net/>

network. The main motivation was to provide customers with enough address space, since solutions like NAT cause problems with voice over IP and peer-to-peer applications. IPv6 connectivity does not introduce any additional costs – this service is charged like IPv4 connectivity.

XS4ALL⁹ is the first broadband provider in the Netherlands that supports IPv6. They also offer native IPv6 access over ADSL without tunneling. The network itself is dual stack and therefore IPv6 ready. If required, XS4ALL provides customers with /48 IPv6 networks, offering a series of 65000 subnetworks to these customers.

There are also other ISPs in Europe that offer an IPv6 access. France Telecom and NTT Europe are two examples. NTT Europe, which is the European arm of NTT Communications (NTT Com), is setting up its own commercial-quality IPv6 POPs in Europe (London, Amsterdam, Paris, Frankfurt and Madrid). This will provide an IPv6 native connection as well as an IPv6 over IPv4 tunneling connection, whereby IPv6 equipment is connected over existing IPv4 networks.

5 Technical Issues

This section describes some important technical issues for an ISP when planing transition from IPv4 to IPv6. These issues must be considered in practical deployment of IPv6. They include Domain Name Service (DNS) transition, routing, security, and network and service operation. We discuss them in the following subsections.

5.1 DNS Transition

Domain Name Service (DNS) has been introduced in IPv4 in order to make addressing user-friendly. It allows using domain names instead of IP addresses. DNS servers resolve domain names on request of end stations into the corresponding IP addresses. To be able to do this, they must maintain bindings between IP addresses and their domain names. These bindings are called *resource records (RR)*. For IPv4 addresses (32 bit long), they are also referred to as *A-records*.

The DNS service is also needed for IPv6. However, IPv6 addresses are 128 bit long, so that A-records cannot be used for storing IPv6 RRs. To solve this problem, IETF has proposed two new RR types:

- AAAA RR, specified in [15], and
- A6 RR, specified in [9].

AAAA was the first proposal. It simply provides the mapping of domain names to 128 bit IPv6 addresses. A6 RRs, in contrast, additionally allow the mapping of IPv6 address prefixes to partial domain names. To resolve a domain name into the corresponding IPv6 address, a DNS server has to resolve a chain of A6 records, each of which can be placed

⁹<https://service.xs4all.nl/>

on a different DNS server. The benefit of this mechanism is the ability to change the address prefix of a domain by changing only a single RR. However, today is the AAAA proposal considered as the industry standard, whereas the A6 proposal as experimental [4].

Although the AAAA RR is just an additional entry to the A RR in a DNS server, adding an AAAA RR for each domain name in a DNS server may not be the optimal solution. In that case, a resource has the same domain name for both A and AAAA RRs, and the application has to decide whether to use A or AAAA RR. Many new applications use automatically an AAAA RR (if one exists). This may lead to degrading performances for those clients whose IPv6 connectivity is not as good as IPv4 one. Note that the DNS query of the application decides which RR to use, and not the connectivity to the DNS server. It is possible to use IPv4 connectivity to obtain an AAAA RR, and vice versa, to obtain an A RR over IPv6 connectivity [40].

The better solution, at least at the beginning of the IPv6 deployment, is to use a separate IPv6 subdomain. In this way IPv4 services use different domain names than IPv6 services, for example *example.com* for A RR and *ipv6.example.com* for AAAA RR. This is very useful at the beginning of IPv6 deployment. Later on, when it is ensured that service level will not degrade for users unaware of IPv6, it make sense to move to the same domain names.

It is also possible to resolve an IPv6 address into the corresponding domain name. This is called *reverse resolution*. In contrast to the forward resolution, which represents an IPv6 address using an AAAA RR, the reverse resolution represents an IPv6 address using a *pointer record (PTR)*. IPv6 PTRs contain the IPv6 address written backwards with dots as separators. They are in the nibble format under the *ipv6.arpa* tree [40]. For transition mechanisms which include a special prefix, the reverse resolution requires a special consideration. Examples of such transition mechanisms are Teredo [17] and 6to4 (see Section 3.2.3). For Teredo it does not seem to be necessary to provide a reverse DNS resolution, given that the IPv6 address bases on the IPv4 address and the port number of the current (short-lived) NAT mapping. For 6to4 several solutions have been proposed. The main proposal [20] describes an autonomous reverse delegation system for 6to4 addresses.

5.2 Routing

Routing protocols discover the network topology and route traffic across the network. In general, there exist two types of routing protocols, Interior Gateway Protocols (IGP) that are used inside a domain and Exterior Gateway Protocols (EGP) that provide the interconnection between adjacent domains. For the successful routing of IPv6 traffic, the routing protocols need to support the new IP version. For dual stack routers, there is a need for a routing protocol for each IP. This can be one routing protocol for both, IPv4 as well as IPv6. Alternatively, it is possible to deploy two separate routing protocols, one for each IP.

Intermediate System-to-Intermediate System (IS-IS), Open Shortest Path First ver-

sion 2 (OSPFv2) and Routing Information Protocol (RIP) are the most common IPv4 internal routing protocols. The most frequently used external routing protocol for IPv4 is the Border Gateway Protocol (BGP) in its current version 4.

	IPv4	IPv6
IGPs	IS-IS, OSPFv2, RIP	IS-IS, OSPFv3, RIPng
EGPs	BGP4, BGP4-MP	BGP4-MP

Table 2: Routing protocol overview

As most of the described protocols were designed to solely support IPv4, they cannot be used for IPv6. Table 2 gives an impression on existing routing protocols for both IP versions. Looking at IGPs, IS-IS supports both, IPv4 and IPv6. OSPF has been rewritten in version 3 to support only IPv6 [8]. RIP for IPv6 is called RIPng [28]. BGP4, which has been the prevalent exterior routing protocol, has been extended for multiprotocol support [7], where IPv4 and IPv6 BGP peerings can coexist. The use of BGP-MP for IPv6 inter-domain routing is specified in [29].

For a migration to dual stack, it is essential to carefully plan the application of the envisaged routing protocols. We recommend to use BGP-MP as EGP as it is already in use by several ISPs. As mentioned earlier, there are two possibilities for the IGP: It is possible to use one routing protocol – which then must be IS-IS – for both IP versions, or to use one routing protocol per IP. The latter approach allows several combinations as illustrated in Table 3. Using one single routing protocol has the advantage of less complexity and higher configuration consistency. The use of two separated protocols really separates the routing processes, i.e. problems occurring with the newer IPv6 do not affect IPv4 (and vice versa). For a further discussion of the routing protocols, please refer to [3].

IPv4	IPv6
IS-IS	OSPFv3
OSPFv2	OSPFv3
OSPFv2	IS-IS
RIP	OSPFv3
RIP	IS-IS
IS-IS	RIPng
OSPFv2	RIPng

Table 3: Examples of IGP combinations

Note that a change of the existing IGP needs to be prepared carefully because the know-how for the new protocol needs to be built in a timely manner.

5.3 Security

In Section 3 we introduced several transition techniques. Each of these techniques has its particular security issues. We will give an overview about security issues with each transition technique and refer the interested reader to the 6net security deliverable [1] for detailed security analysis. We will start our security investigation with dual stack concerns, proceed with tunnel issues and conclude with translation topics.

The most important thing to notice is that each security rule on a dual stack host or router has to be deployed twice – once for IPv4 and once for IPv6. It is important that these rules are consistent, as a missing rule may provide an open door for potential attackers. An interesting issue is that dual stack hosts behind a NAT that do not have IPv4 connectivity may be potential targets, as these hosts may be accessible via IPv6 unique addresses and a tunnel mechanism. A replication of IPv4 firewall rules for IPv6 traffic filtering should eliminate this danger.

As tunneling itself is a security threat to a network, it is very critical in security respects. Each tunnel circumvents fine-grained security access control lists established on the domain border. The security analyzer residing at a network edge device might only inspect the traffic's outer layer, while the content – the (maybe harmful) tunneled packets – arrives at the tunnel exit point behind the security border. Once decapsulated, the traffic may do much damage, as the main defense has been conquered. IP tunnels during network migration provide similar security holes. Especially IPv4 firewalls that are not IPv6 aware, cannot enforce any filter operation on tunneled IPv6 traffic. For IPv6 traffic encapsulated in IPv4, these firewalls can just permit or deny IPv6 tunnels (IP protocol type 41 payload). This means that either there is no IPv6 connectivity for the network protected by this firewall or the network is totally open for this tunneled IPv6 traffic. Solutions provide only IPv6 capable firewalls that may be collocated with the tunnel exit point. This allows a fine grained control over the IPv6 traffic that enters the network.

Translation mechanisms can occur either at network, transport or application layer and usually base on an intermediate server that performs the translation. Apparently, this translator is a man-in-the-middle (MITM), which always breaks end-to-end security, i.e. IPSec. As this MITM hides the real source address of its clients, it may provide users unauthorized access to services or resources. Finally, because of their logistic importance for many users, these translators are a very interesting target for Denial of Service (DoS) attacks.

5.4 Network and Service Operation

In an IP network, there are many entities that require managing and monitoring. These tasks not only improve operation, management and security of a network, but also give an important input for research and development. They fall into responsibility of the network and service operation (see Figure 18). For an IPv6 network, network and service operation accomplishes the following tasks [26]:

- Establishment of IPv6 connectivity to other ISPs and peers,
- IPv6 configuration of network devices,
- IPv6 network management,
- IPv6 network monitoring,
- IPv6 customer management, and
- IPv6 network and service operation security.

In this section, we discuss the influence of the IPv4-IPv6 transition on the network management. In the IPv4 world, the standard protocol for network management is the Simple Network Management Protocol (SNMP) [6]. It is an application layer protocol used to exchange management information between network devices. The management information of devices are stored in *Management Information Bases (MIBs)*. The *Network Management System (NMS)* is able to access these MIBs over SNMP. Figure 22 shows an example of a SNMP managed network.

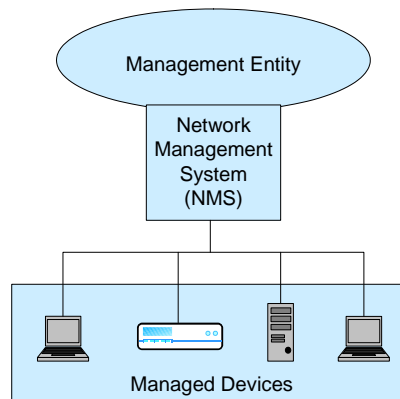


Figure 22: A SNMP managed network

The smooth integration of IPv6 into network management is a very important factor for deployment of IPv6. As a first step, all network devices can be managed over the IPv4 transport. This is possible because IPv6 MIBs can be collected over the IPv4 transport. As a second step, management information can be exchanged over IPv6. A benefit of this is the possibility to use network monitoring functions that require IPv6 transport. An example is a monitoring application that uses ICMPv6 messages. There are several requirements for this second step:

- IPv6 support on NMS,
- IPv6 support on manageable devices,
- SNMP over IPv6, and

- IPv6 address support on MIBs when required.

We expect that both IPv4 and IPv6 will co-exist for several years. During this time, both protocol versions have to be managed. This is possible in this second step, since IPv4 MIBs can be transported over IPv6.

6 Summary and Conclusion

This deliverable examined mechanisms and strategies for transition from IPv4 to IPv6, with the particular focus on large ISP networks. First, we pointed out the benefits of IPv6 compared with IPv4. Although the IPv4 address exhausting has been postponed by different techniques, it is still a question of time. Furthermore, a lot of functionality supported in IPv4 by different extensions are natively supported in IPv6. Therefore, it is important for ISPs to consider strategies for a soft transition to IPv6.

We have described several existing transition mechanism and presented the possible transition stages for an ISP. However, this document described the transition from an IPv4 to a dual stack ISP network. Later on, an ISP might want to migrate to an IPv6 only network, at least on some parts of its network. In fact, providing IPv6 only services does not differ a lot from providing dual IPv4 and IPv6 services. The difference is that IPv4 services must be maintained over an IPv6 only network in this case.

Concluding, we recommend an IPv4-IPv6 transition starting on provider access networks, and using a tunneling technique in the backbone. This provides customers with IPv6 support, without a major upgrade of the core infrastructure and without an impact on existing IPv4 services. This allows a cost-efficient evaluation of IPv6 products and services before a full IPv6 deployment in the network. Later on, the whole network infrastructure can be upgraded to natively support IPv6, either in the dual-stack manner or even in the IPv6 only manner. Interconnection with other IPv6 ISPs and with the 6bone ensures better understanding of IPv6 requirements and additional assessment and evaluation of IPv6.

References

- [1] G. D. Battista, B. Tuy, L. Colitti, J. Durand, R. Evans, D. Kalogeras, G. Koutepas, M. Patrignani, P. Savola, S. Venaas, C. Strauf, T. Strauf, T. Kersting, and S. Leinen. Operational procedures for secured management with transition mechanisms (version 2). Report 6.2.2v2, 6net, May 2004.
- [2] M. Blanchet and F. Parent. IPv6 Tunnel Broker with the Tunnel Setup Protocol (TSP). Internet draft, work in progress, Internet Engineering Task Force, 2004.
- [3] E.-J. Bos, N. den Otter, P. Savola, G. Paolini, V. R. Carcione, J. Durand, L. Lhotka, C. Schild, R. Evans, D. Rogerson, R. Samani, D. Kalogeras, S. Venaas, T. Skjesol, F. Karayannis, A. Liakopoulos, C. Tziouvaras, J. Mohacsi, S. Leinen, C. Friacas, J. P. Sorensen, W. Woeber, and B. Gajda. Updated IPv4 to IPv6 transition Cookbook for organisational/ISP (NREN) and backbone networks. Report 2.2.3, 6net, May 2004.
- [4] R. Bush, A. Durand, B. Fink, O. Gudmundsson, and T. Hain. Representing Internet Protocol version 6 (IPv6) Addresses in the Domain Name System (DNS). RFC 3363, IETF, August 2002.
- [5] B. E. Carpenter and K. Moore. Connection of IPv6 Domains via IPv4 Clouds. RFC 3056, IETF, February 2001.
- [6] J. Case, R. Mundy, D. Partain, and B. Stewart. Introduction to Version 3 of the Internet-standard Network Management Framework. RFC 2570, IETF, April 1999.
- [7] T. B. R. Chandra, D. Katz, and Y. Rekhter. Multiprotocol Extensions for BGP-4. RFC 2283, IETF, June 2000.
- [8] R. Coltun, D. Ferguson, and J. Moy. OSPF for IPv6. RFC 2740, IETF, December 1999.
- [9] M. Crawford and C. Huitema. DNS Extensions to Support IPv6 Address Aggregation and Renumbering. RFC 2874, IETF, July 2000.
- [10] J. Davies, C. Huitema, S. Tansley, M. Talwar, and D. Thaler. Torero Overview. Microsoft TechNet Article, Microsoft Corporation, <http://www.microsoft.com/technet/prodtechnol/winxppro/maintain/teredo.msp>, July 2004.
- [11] S. E. Deering and R. M. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460, IETF, December 1998.
- [12] A. Durand, P. Fasano, I. Guardini, and D. Lento. IPv6 Tunnel Broker. RFC 3053, IETF, January 2001.

- [13] P. Ferguson and D. Senie. Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. RFC 2827, IETF, May 2000.
- [14] R. E. Gilligan and E. Nordmark. Transition Mechanisms for IPv6 Hosts and Routers. RFC 2893, IETF, August 2000.
- [15] R. M. Hinden and S. E. Deering. IP Version 6 Addressing Architecture. RFC 2373, IETF, July 1998.
- [16] C. Huitema. An Anycast Prefix for 6to4 Relay Routers. RFC 3068, IETF, June 2001.
- [17] C. Huitema. Teredo: Tunneling IPv6 over UDP through NATs. Internet Draft, Internet Engineering Task Force, 2004.
- [18] C. Huitema, R. Austein, S. Satapati, and R. van der Pol. Evaluation of IPv6 Transition Mechanisms for Unmanaged Networks. Internet Draft, Internet Engineering Task Force, 2004.
- [19] C. Huitema, R. Austein, S. Satapati, and R. van der Pol. Unmanaged Networks IPv6 Transition Scenarios. RFC 3750, Internet Engineering Task Force, 2004.
- [20] G. Huston. 6to4 Reverse DNS Delegation. Internet draft, work in progress, Internet Engineering Task Force, April 2004.
- [21] J. ichiro itojun Hagino and K. Yamamoto. An IPv6-to-IPv4 Transport Relay Translator. RFC 3142, IETF, June 2001.
- [22] C. Kawa, A. G. M. R. Cherukuri, D. Sinicrope, P. Pate, R. Bhat, N. Vasavada, L. Martini, N. El-Aawar, G. Heron, D. S. Vlachos, D. Tappan, E. Rosen, S. Vogel-sang, V. Sirkay, C. Liljenstolpe, and K. Kompella. Frame Relay over Pseudo-Wires. Internet draft, work in progress, IETF, August 2004.
- [23] H. Kitamura. A SOCKS-based IPv6/IPv4 Gateway Mechanism. RFC 3089, IETF, April 2001.
- [24] L. Ladid. IPv6 on everything: the new Internet IPv6 helps network architects address the IP address shortage, security, QoS, multicast and management. In *Proceedings of the 3G Mobile Communication Technologies Conference*, pages 317–322, 2001.
- [25] M. Leech, M. Ganis, Y.-D. Lee, R. Kuris, D. Koblas, and LaMont. SOCKS Protocol Version 5. RFC 1928, IETF, March 1996.
- [26] M. Lind, V. Ksinant, S. D. Park, A. Baudot, and P. Savola. Scenarios and Analysis for Introducing IPv6 into ISP Networks. Internet Draft, Internet Engineering Task Force, 2004.

- [27] A. G. Malis, J. Brayley, S. Vogelsang, J. Shirron, and L. Martini. SONET/SDH Circuit Emulation Service Over MPLS (CEM) Encapsulation. Internet draft, work in progress, IETF, May 2004.
- [28] G. S. Malkin and R. E. Minnear. RIPng for IPv6. RFC 2080, IETF, January 97.
- [29] P. R. Marques and F. Dupont. Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing. RFC 2545, IETF, March 1999.
- [30] L. Martini, M. Bocci, N. El-Aawar, J. Brayley, G. Koleyoni, G. Heron, D. S. Vlachos, D. Tappan, J. Jayakumar, E. C. Rosen, S. Vogelsang, G. de Grace, J. Shirron, A. G. Malis, V. Sirkay, C. Liljenstolpe, K. Kompella, J. Fischer, M. Aissaoui, T. Walsh, J. Rutemiller, R. Wilder, and L. Dominik. Encapsulation Methods for Transport of ATM Over MPLS Networks. Internet draft, work in progress, IETF, October 2004.
- [31] L. Martini, N. El-Aawar, G. Heron, E. C. Rosen, A. G. Malis, D. Tappan, S. Vogelsang, V. Sirkay, V. Radoaca, C. Liljenstolpe, K. Kompella, T. So, X. Xiao, C. O. Flores, D. Zelig, R. Sharma, N. Tingle, S. Khandekar, and L. Andersson. Encapsulation Methods for Transport of Ethernet Frames Over IP/MPLS Networks. Internet draft, work in progress, IETF, October 2004.
- [32] L. Martini, N. El-Aawar, G. Heron, D. S. Vlachos, D. Tappan, J. Jayakumar, A. Hamilton, E. Rosen, S. Vogelsang, J. Shirron, T. Smith, A. G. Malis, V. Sirkay, V. Radoaca, C. Liljenstolpe, D. Cooper, and K. Kompella. Encapsulation Methods for Transport of Layer 2 Frames Over IP and MPLS Networks. Internet draft, work in progress, IETF, September 2004.
- [33] L. Martini, N. El-Aawar, G. Heron, D. S. Vlachos, D. Tappan, J. Jayakumar, A. Hamilton, E. Rosen, S. Vogelsang, J. Shirron, T. Smith, A. G. Malis, V. Sirkay, V. Radoaca, C. Liljenstolpe, D. Cooper, and K. Kompella. Transport of Layer 2 Frames Over MPLS. Internet draft, work in progress, IETF, June 2004.
- [34] E. Nordmark. Stateless IP/ICMP Translation Algorithm (SIIT). RFC 2765, IETF, February 2000.
- [35] D. Ooms, J. D. Clercq, S. Prevost, and F. L. Faucheur. Connecting IPv6 Islands over IPv4 MPLS using IPv6 Provider Edge Routers (6PE). Internet draft, work in progress, IETF, April 2004.
- [36] J. Postel. Internet Protocol. RFC 791, IETF, September 1981.
- [37] E. C. Rosen, D. Tappan, Y. Rekhter, G. Fedorkow, D. Farinacci, T. Li, , and A. Conta. MPLS Label Stack Encoding. RFC 3032, IETF, Jan 2001.
- [38] E. C. Rosen, A. Viswanathan, and R. Callon. Multiprotocol Label Switching Architecture. RFC 3031, IETF, January 2001.

- [39] A. S. Tanenbaum. *Computer Networks, Third Edition*. Prentice Hall Inc., 1996.
- [40] S. Thomson, C. Huitema, V. Ksinant, and M. Souissi. DNS Extensions to Support IP Version 6. RFC 3596, IETF, October 2003.
- [41] G. Tsirtsis and P. Srisuresh. Network Address Translation - Protocol Translation (NAT-PT). RFC 2766, IETF, February 2000.
- [42] D. G. Waddington and F. Chang. Realizing the Transition to IPv6. *IEEE Communications Magazine*, 6(3):138–148, 2002.