Automatic IPv4 to IPv6 Transition D1.1 - Network Topologies and Transition Procedures

Frédéric Beck, Isabelle Chrisment, Olivier Festor

Automatic IPv4 to IPv6 Transition D1.1 - Network Topologies and Transition Procedures

Frederic Beck, Isabelle Chrisment, Olivier Festor

June 5, 2009

Contents

Abstract

Over the last decade, IPv6 has established itself as the most mature network protocol for the future Internet. While its acceptance and deployment remained so far often limited to academic networks, its recent deployment in both core networks of operators (often for management purposes) and its availability to end customers of large ISPs demonstrates its deployment from the inside of the network leading to the edges.

For many enterprises, the transition remains an issue today. This remains a tedious and error prone task for network administrators.

In the context of the Cisco CCRI project, we aim at providing the necessary algorithms and tools to enable this transition to become automatic. In this report, we present the first outcome of this work, namely an analysis of the transition procedure and a model of target networks on which our automatic approach will be experimented. We also present a first version of a set of transition algorithms that will be refined through the study.

Chapter 1

Introduction

IP networks are widely spread and used in many different applications and domains. Their growth continues at an amazing rate sustained by its high penetration in both the Home networks and the mobile markets. Although often postponed thanks to hacks like NAT, the exhaustion of available addresses, and other scale issues like routing tables explosion will occur in a near future.

The IPv6 [5] was defined with a bigger address space (128 bits) and comes along with new built-in services (address autoconfiguration [18], native IPSec, routes aggregation, simplified structure...). It is a fact that IPv6 deployment is slower than foreseen. Many reasons are valid to explain this: economical, political, technological, and human. Despite this slow start, IPv6 is today more than ever the most mature network protocol for the future Internet. To faster its acceptance and deployment however, it has to offer autonomic capacity that emerge in several recent protocols in terms of self-x functions reducing end often eliminating human intervention in the loop. We are convinced that such features are also required for the evolutionary aspects of an IP network, the transition from IPv4 to IPv6 being an essential one.

In this project, we are interested in the scientific part of the technological problems that highly impact human acceptance. Many network administrators are indeed reluctant to deploys IPv6 because, first, they do not know well the protocol itself, and they do not have sufficiently rich algorithmic support to seamlessly manage the transition from their IPv4 networks to IPv6. To address this issue, we investigate, design and aim at implementing a transition framework with the objective of making it self-managed.

As the IPv4 to IPv6 transition is a very complex operation, and can literally lead to the death of the network, there is a real need for a transition engine to ease and secure the network administrator's task; the ideal being a "one click" transition.

This report presents the first step of the work, in which we lay the foundation stones of our work. First, we present the transition, its mechanisms and the problems it raises. In section 3, we will present the different topologies we will address during our study, and which will be the infrastructure on which we will run all our simulations and experiments. Finally, we give a first version of a Transition Procedure. This procedure will evolve and be amended during the whole study. The objective at the end of the study is to have a safe procedure to help the administrators to deploy IPv6 on their network.

This report is not meant to be exhaustive, as we did not present all existing transtion mechanisms (we did not mention 6rd [6] which is used by a French ISP, or TEREDO). It is used to set the basis of our study, to make our ideas clear on the goals we are aiming to reach and the problems we want to address. Later reports will extend some parts of this reports, like D1.2 will extend and details the topologies we will consider. In D1.2, we will take one by one the topologies detailed in this report and take a closer look on the possible issues they can raise. Moreover, even if we presented different transition mechanisms, we will focus during our study, and thus this report, on the Dual Stack transition, which we believe will be the most usual scenario.

Chapter 2

IPv4 to IPv6 Transition

2.1 Transition Mechanisms

In this section, we briefly present the different existing transition mechanisms. They can be separated in 3 different categories as presented in [9]:

- Dual IP layer operations,
- Tunneling mechanisms,
- Translation mechanisms.

2.1.1 Dual IP Layer operations

Dual IP layer operations is the most straightforward way for IPv6 nodes to remain compatible with IPv4 only nodes. These nodes can exchange IPv6 packets with IPv6 nodes, and IPv4 packets with IPv4 only nodes, as both type of addresses are configured on the host. When using dual stack, hosts can choose to desactivate either the IPv6 stack and become IPv4 only hosts, or desactivate the IPv4 stack to become IPv6 only hosts. The addressing mechanism used to configure the IPv4 and IPv6 addresses are independent.

Moreover, these hosts have two types of DNS entries: A type records for the IPv4 address, and one or more A6 (or AAAA) record(s) for their IPv6 address(es). But before advertising this AAAA record, 3 conditions must be fulfilled:

1. the address is assigned to the interface on the node,
2. the address is configured on the interface,
3. the interface is on a link which is connected to the IPv6 infrastructure.

These hosts will prefer to use IPv6 rather than IPv4 when possible. They will try to resolve first in IPv6, and will switch to IPv4 only if the IPv6 resolution or connection fails, which may add delay in the users requests in case of errors.

2.1.2 Tunneling Mechanisms

This method is used to make IPv6 islands communicate via the IPv4 network, by encapsulating IPv6 packets into IPv4 packets. To do so, tunnels are set between two IPv6 islands. The endpoints of these tunnels can be routers or hosts. A network prefix is generally assigned to the network managed by a router endpoint, whereas a single IPv6 address is assigned for a single host.

The mechanisms are pretty well known, such as 6over4 [2] where the hosts use IPv4 multicast, or 6-to-4 [3], usually used for interconnecting an IPv6 island to the IPv6 backbone via an unicast tunnel to a 6-to-4 router with takes care of the encapsulation/decapsulation of the packets.

When connecting a single host to the IPv6 backbone and not a whole network, we use Tunnel Brokers. The are two categories of tunnel brokers, configured tunnels and automatic tunnels.

Configured Tunneling

In configured tunneling, the tunnel endpoint address is determined from configuration information. This address is used as the destination address for the encapsulating IPv4 header. At least the endpoints must have both an IPv4 and IPv6 address.

Default Configured Tunnel

IPv4/IPv6 hosts that are connected to a link with no IPv6 capable router configure a static tunnel to an IPv6 capable router (e.g. 6to4).

Default Configured Tunnel using IPv4 "Anycast Address"

In the same way, with this type of tunnel, a static tunnel is set up, but instead of using an unicast IPv4 as destination endpoint, an anycast address is used. This provides a better robustness, as multiple routers can be reached, and it benefits from the usual fall-back mechanisms of IPv4 routing.

Automatic Tunneling

A host or router uses the IPv4 compatible destination address of the IPv6 packet being tunneled.

2.1.3 Translation Mechanisms

The translation mechanism is used when IPv6 only nodes want to communicate with IPv4 only nodes and no IPv6 in IPv4 mechanism is available. Different mechanisms exist for this case:

- Translation of the IPv4 header into an IPv6 header, and vice versa (SIIT [17], NAT-PT [19]
- Bump in the stack [20], where SIIT or NAT-PT operations are performed directly into the stack, which permits to give transparently access to IPv6 to IPv4 only applications
- Bump in the API [13], where the headers are not translated, but the sockets are
- Transport Relay, where IPv4 and IPv6 hosts exchange TCP and UDP packets via transport relay translator [10]
- SOCKS [14, 12] which acts as an IPv4/IPv6 proxy
- Application Layer Gateways such as squid or apache HTTP proxies

2.2 Transition Components and Prerequisites

The components to take into accounts when doing a transition are not limited to hosts and network components. In this section, we will identify all the components of such an operation, and highlight the prerequisites linked to them.

These components can be IPv4 only, IPv6 only, or use both protocols at the same time (see section 2.1 for more details).

Hosts

Depending on the transition mechanism chosen, the hosts must be able to adapt itself to the new network. IPv4 only hosts must not be influenced by the IPv6 network. IPv6 only hosts must be able to communicate with the IPv6 world, which means that they must be IPv6 compatible (IPv6 activated in the operating system) and that they must have IPv6 compatible applications (WEB browser, Mail client...). Finally, hosts using both protocols cumulate the constraints of both IPv4 and IPv6 hosts.

Network Components and Routing Protocols

From the network point of view, network components (routers, switches...) must follow the same rules than the hosts. The components meant to use IPv6 must have the appropriate firmware that enables this feature, the IPv6 routing infrastructure should not interfere with the IPv4 one.

IPv6 routing protocols are fully operational, and for most of them are simply new versions of the existing IPv4 ones. Thus, it will be easy to choose one of them, as the same advantages and drawbacks than in IPv4 have to be considered. The only prerequisite is that the network components do have the chosen protocol in their feature list.

Firewalls

Firewalls and all security related components have to be ready for the transition, which means that IPv6 enabled components must be present, at least a firewall in the first time, to avoid creating vulnerabilities that could compromise both IPv4 and IPv6 networks. It is important to secure the IPv6 network before enabling it.

DNS

During the transition phase where hosts have both IPv4 and IPv6 addresses, the DNS should respond to a resolution with both the v4 and v6 addresses. That implies that the AAAA records have been added into the name server. However, it is important to not add the v6 entry in the DNS as long as the corresponding host is not IPv6 capable, to avoid black holes.

As IPv6 is the protocol used by default, in case of failure for a given service, a host that performs a name resolution should switch quickly enough to the IPv4 service to avoid that the end-users notice a slowing down in their network access.

Services

The last important components are the services. Having an IPv6 capable network and host is a first step, but if the services and applications are not IPv6 capable, the transition will seem useless. Many services are already available for IPv6, like HTTP servers (Apache...), Mail servers, management plane... But some other specific applications may need a portage to the new socket API or are simply not yet available. It is important to check the availability of the services running on a network before migrating it to IPv6, and depending on the result of this study, chose the appropriate transition mechanisms.

2.3 Key Transition Elements

In our study, we will not discuss whether IPv4 or IPv6 is the best choice, if IPv6 is required, if IPv4 will disappear one day... We strongly believe that IPv6 will be an important part in the Future Internet, and we will focus on how it can be safely and efficiently deployed on existing IPv4 network. This operation is called IPv4 to IPv6 transition. We saw in section 2.1 that the operation called transition does not always mean that the network will run only IPv6, but that in most cases the network (or at least some components) use both protocols at the same time.

When transitioning from IPv4 to IPv6, the key is to maintain compatibility between the IPv4 and IPv6 networks, and avoid having 2 completely separated infrastructures. If done properly, it will permit to IPv4 only and IPv6 only hosts to communicate, and will ensure a smooth migration from the IPv4 Internet to an IPv6 Internet.

To reach that goal, the first step is to identify the key elements of the transition [1], answer the questions raised by each of them, and try to automate everything that can be.

2.3.1 Identify Network Infrastructure

Before performing the transition, it is mandatory to review the whole network infrastructure, and thus identify all components that will play a role in the transition. Routers, switches, firewalls... must be checked closely, and eventually upgraded to support IPv6.

Can this part be automated ? In what extend ?

2.3.2 Identify an Addressing Plan and Request Addresses

Prior to requesting IPv6 addresses, each organization should determine the required IPv6 address space, to make sure the prefix requested will be sufficient, and will support extensions in a near future. Once this is done, the organization can request its network prefix.

Then, the organization will have to define the network topology and hierarchy of their network, in order to set up an addressing plan while respecting prefix aggregation. We believe that this function can be automated, and we will integrate it in our approach.

2.3.3 Identify Transition Mechanisms

As presented in section 2.1, different transition mechanisms exist. These mechanisms are intended to ensure interoperability between IPv4 and IPv6. Having already identified the network infrastructure and set an addressing plan, an organization can identify the best suited mechanisms to its needs. It should be noted that, depending on its needs, an organization can choose to use multiple transition mechanisms.

Can we automate this function ?

2.3.4 Identify Security Plan

When performing the transition, it is mandatory to avoid the creation of new security holes that could endanger the IPv6 or the existing IPv4 network, as the transition does not compromise the existing IPv4 network.

What are the parts that can be automated, and how ?

The security applications infrastructure currently used on an IPv4 network will need to be replicated, with an expectation that the same level of assurance is provided in the IPv6 network. The security policy of the organization will need to be adapted to the specificities of IPv6. This includes intrusion detection systems, network management, virus detection, secure web functions... A closer look should be given to IPSec, VPN, Mobile IPv6 [11] and wireless if implemented on the transitioned network.

Depending on the transition mechanism used (e.g. IPSec tunnel used to encapsulate IPv6 within IPv4), some specific (and temporary) measures have to be taken to ensure a secure transition.

But, whatever the case is, IPv6 should not be enabled on a network unless all network security infrastructures are implemented, especially ingress filtering.

2.4 Issues Raised

During a transition operation, several issues can or do arise.

[1] `http://www.cio.gov/documents/IPv6\_Transition\_Guidance.doc`

2.4.1 Non Technical Issues

The non technical issues are impossible to automate.

Cost

The cost of the transition appears after the network infrastructure and the addressing plan have been defined. Depending on the cases and needs, this cost can grow quickly, and thus be a slowing down in IPv6 deployment.

When provisioning a transition, it is important to take into account the cost of the IPv6 ISP (network connection + prefix), eventually new hardware or software updates, training needs, men power... The transition may be the right moment to update an old and deprecated network into a more efficient one.

Maybe the evaluation function can be automated, or at least partially.

Training Needs

Training will be an important part of the integration process of IPv6. The staff (network administrators, support team...) will potentially need to be specially educated and trained for IPv6 and the new features that it will bring into the network, or for the new hardware that comes along with it. The specific cost of training each person will depend upon its role in the IT infrastructure, and the knowledge of each one.

Human Factor

The human factor can also be considered as an issue, as network administrators are usually reluctant to modify their network, and integrate IPv6 (and generally anything new) in a working infrastructure. They may not be easy to convince, as the temporary mechanisms set up to postpone the address exhaustion are sufficient for them. This reluctance may lead to an increase in the cost of the training, as the administrators may need more training to get reassured and feel confident enough.

2.4.2 DNS

DNS is one of the big issues when performing a transition. Besides the fact that the AAAA records must be added in the name server, there are two main issues.

First, it is important to follow the guidelines announced in section 2.1.1 before advertising the new addresses assigned to the hosts in the transitioned network.

Moreover, as IPv6 is the first choice when available, it is important to avoid to announce inaccessible IPv6 addresses, and to be able to quickly switch to IPv4 in case of troubles with IPv6, so that the users do not feel harm by the transition.

2.4.3 Routing Related Issues

IPv4 to IPv6 transition results in a dual IP layer transition, augmented by the use of encapsulation or translation where it is necessary or appropriate. Routing issues related to this transition are described in [1]. It concerns:

- the routing of IPv4 packets
- the routing of IPv6 packets, with IPv6 native or IPv4 compatible addresses
- the operation of manually configured tunnels
- the operation of automatic encapsulation, by locating the encapsulators and ensuring that the routing is consistent with the encapsulation

The basic mechanisms for routing both IPv4 and IPv6 involves a dual IP layer routing where IPv4 and IPv6 routes are separately calculated and processes. This does not mean that the routing should be separated on different network components, but that each stack computes its own routes.

Tunnels are treated as if they were normal links. This is the case for static tunnels, but also for automatic ones. But as this second category uses IPv4 anycast addresses, consistency between both IPv4 and IPv6 routing is required, especially when performing recovery procedures if the endpoint of the tunnel breaks.

Chapter 3

Dual Stack Transition and Topologies

3.1 Considered Situation

In our study, we will focus on the addressing plan and the configuration of the network. We will not consider the scenario in which a single host establishes a tunnel to get IPv6 access. We will not consider either how the network is connected to the IPv6 backbone. Many tutorials and documentation already exist, explaining how to set up a tunnel, or get a native IPv6 connection. We will make the assumption that the border router is already connected to the IPv6 world.

We will consider that the transitioned network will be in a dual IP layer situation. We believe that it will be the most commonly used case, as it ensures the best interoperability between IPv4 and IPv6, and offers the possibility to desactivate one of the protocols on a given subnet for any reason. Thus, it ensures a smooth and progressive transition, which will be transparent for the users while offering the best service possible for these end users.

Section 3.2 shows within that scenario the topologies we address during this study.

3.2 Considered Topologies

In this section, we will present the different topologies we will address during this study. They represent different cases, from the most simple one to more complex ones. Each topology has its own specificity that can make the transition harder or raise specific issues. For each topology, we will adapt the algorithms we will define to integrate these specificities.

This section will be extended and discussed in details in D1.2.

3.2.1 Tree

This is the most simple case, where the network can be represented as a tree, as shown in figure 3.1.

The routers and subnets are expressed through a hierarchical organization, on which the algorithm will base itself for route aggregation.

This topology will be used to define the first version of the algorithm, in which we will have a minimal set of constraints. All the subnet do perform address autoconfiguration, the only constraint being that some routers will need to handle more subnets in a near future, and need a reserved pool of available prefixes.

Taking this reserved pool into account, the algorithm must propose an addressing plan for the network respecting prefix aggregation.

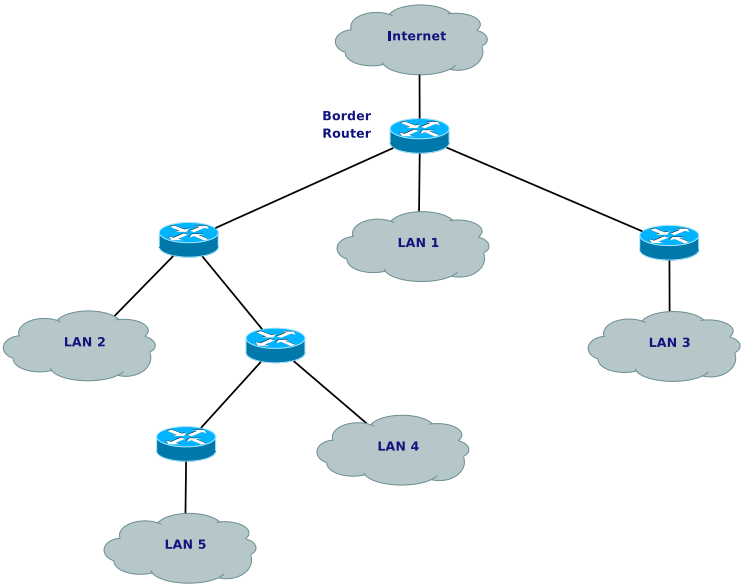

Figure 3.1: Simple Network as a Tree

3.2.2 DMZ

In this case, the network is still represented as a hierarchical tree, but the network is also composed of a DMZ, as shown in figure 3.2.

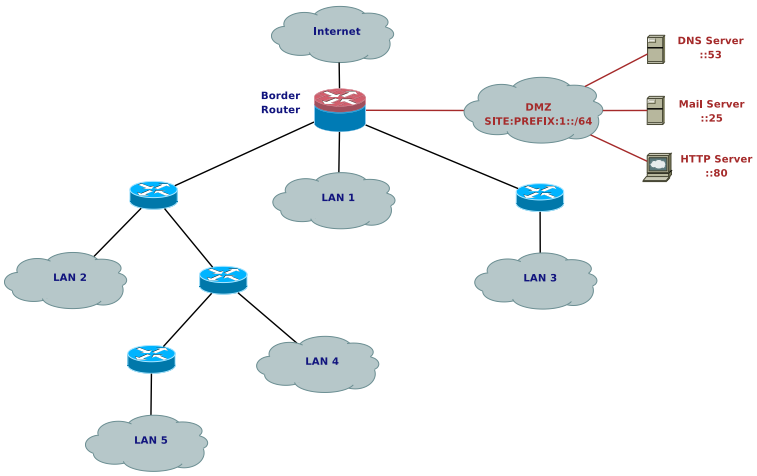

Figure 3.2: Network with a DMZ

The DMZ is a particular subnet, with specific constraints, in terms of security, addressing... In this situation, one of the keys is the update of the firewall rules to protect the DMZ.

A typical constraint is that the DMZ does not perform address autoconfiguration, but uses static addressing or DHCPv6. Another one would be to force a given network prefix on the DMZ or any other subnet.

3.2.3 Loop

We still keep the same basis, but we introduce a loop in the routing, with one subnet being connected to two routers at the same time, for redundancy issues, as we can see in figure 3.3.

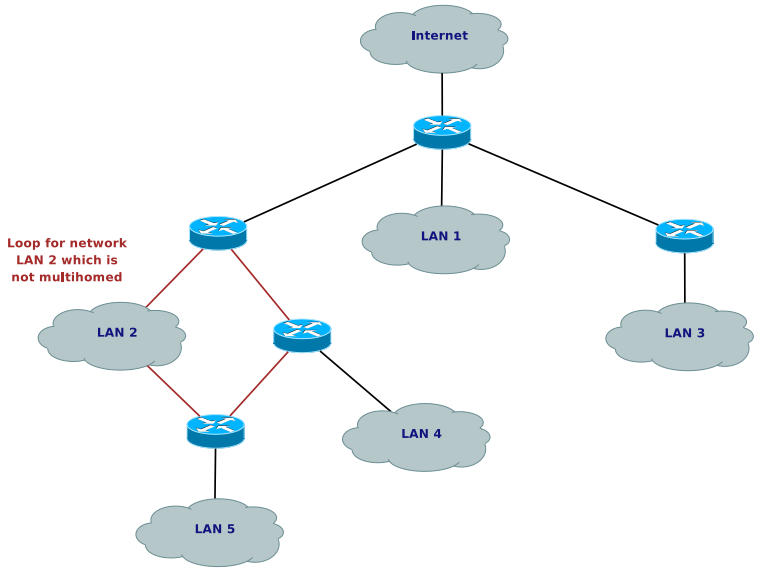

Figure 3.3: Network with a Loop

The difficulty here is to define which router will be the primary upstream for the subnet, and which one will be the backup. Moreover, the subnet could be multihomed with both routers advertising a different prefix. The constraints added in the algorithm would be that the subnet could be multihomed or not, and the possibility to force the choice of the upstream router. Otherwise, the algorithm should determine which router will be used as the main upstream. Some load type metrics can also be used to determine that upstream. Priorities based on the IPv4 routing protocol can also be used to designate the upstream router and its backup.

This topology highlights the fact that some information can be deduced from the IPv4 network.

3.2.4 VLAN

This time, the specificity is that VLANs have been deployed on the network as seen in figure 3.4.

Usually, a single router, the border router, takes care of the routing between the different VLANs. The other equipments on the network are switches whose role is just to dispatch the VLANs. However, some of the switches can perform inter-VLAN routing as well, or can be replaced or coupled to routers.

The VLANs are spread over the whole network, and the addressing plan should respect the definition of these VLANs, while trying to respect as much as possible prefix aggregation.

As such an architecture can end up being very complicated, we will focus as a primary step in the situation described in figure 3.4, where one router is in charge of the routing, and the switches dispatch the VLANs on their ports. We will consider more complex cases if we have time left at the end of the study.

3.2.5 Different Border Router

In this case, we assume the border routers managing IPv4 and IPv6 are different, as shown in figure 3.5.

It implies that some network are IPv4 only, whereas others are IPv6 only.

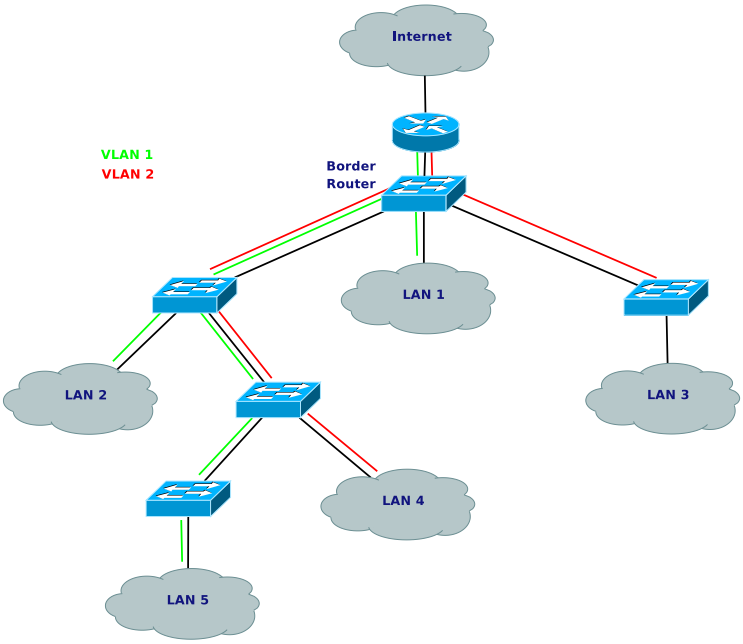

Figure 3.4: Network with VLANs

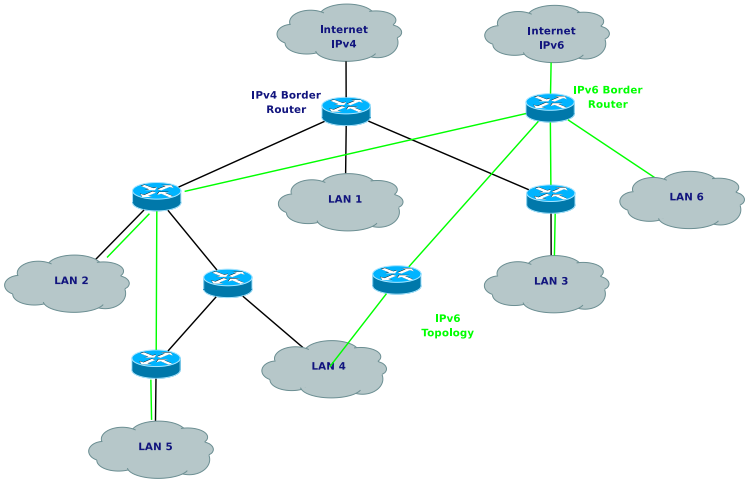

Figure 3.5: Network with Different IPv4 and IPv6 Border Routers

3.2.6 Merging

This final case represents the outcome of the study, where the algorithm should run on all these topologies and integrate all the constraints related to them, and the ones that will be defined independently from the topologies specificities.

Another scenario that could be taken into account, merging specificities of different topologies, if the multihoming of the whole site or a part of it for redundancy issues (for example the DMZ), either with two different ISP (on the same border router or not), or with the same ISP but with two different border router, one being the main upstream, the other one being the backup one. This would imply to run the algorithm twice, while taking the results of the first pass as constraints for the second one.

We aim at providing a set of constraints that the algorithm will be able to take as input, and identify which one will have to be defined by the network administrator, and which ones can be deduced by the existing IPv4 network, or the management plane.

Chapter 4

Transition Procedure

In this chapter, we present a first version of a transition procedure, that will be the starting point of our study, leading to the definition of guidelines and implementation of tools enabling a smooth transition. We will not take into account in that procedure the actions that can not be automated.

4.1 Procedure

1. Identify the network infrastructure
2. Identify the needs in term of addressing and request site prefix
3. Set ingress filtering
4. Connect the border router to the IPv6 world
5. Identify the constraints
6. Determine an addressing plan
7. Define and set firewall rules
8. Update DNS entries
9. Configure routing infrastructure and address routers
10. Address nodes
11. Verify addressing of core services
12. Advertise the prefix and DNS

We will now detail each step of that procedure, and highlight some open issues that we will address during our study.

4.1.1 Identify the network infrastructure

The first step when planing a transition, it to make an audit of the network infrastructure in order to identify the network topology and components. All the components that will take part in the transition must be IPv6 capable. If it is not the case, they must be updated or upgraded, unless these parts of the network are meant to stay IPv4 only.

This operation can be eased by the existing management plane. As administrators deploy applications to ease the management of a network, most of the informations required at this step can be extracted

from these (and especially CiscoWorks). However, as there are many different applications, even specific home made ones, this raises several questions:

- What are the applications usually deployed in the management plane in real networks ?
- What are the information we can extract from it ?
- What about CiscoWorks ? Is it possible to implement the transition engine as a plugin via its API ? If not, can we extract information about the topology and network components from it ? How ?

4.1.2 Identify the needs in term of addressing and request site prefix

Once the network architecture is defined, the next step is to identify the needs, in order to determine the length of the network prefix to ask for. Each subnet or link needs at least one /64. Moreover, it is necessary to plan short or mid term growth of the network and possible appearance of new subnets, in order not to ask for a too short prefix.

Then, simply request the prefix to the ISP or authoritative organization.

4.1.3 Set ingress filtering

Once a prefix has been assigned to the site, the first thing to do is to set ingress filtering, by enabling ACLs on the border firewall before doing any configuration operation on the network. By doing that, we ensure that we do not create vulnerabilities when we begin to enable and deploy IPv6 on the network.

It is important not to advertise the prefix outside the network at that step, as nothing is configured yet.

4.1.4 Connect the border router to the IPv6 world

This operation is already well documented, many different tutorials exist, detailing the methods and different mechanism that can be used (native connection, tunnel or translation, see section 2.1).

As we are not focusing on this operation, and make the assumption that the border router is already connected, we will not detail it. We chose to put it in that position, to highlight the fact that before doing any IPv6 operation at the network scope, the ingress filtering should be set, but this connection can be set at any moment, even after the nodes are addressed.

4.1.5 Identify the constraints

Now, we have to identify the constraints and specificities of the network in order to define the addressing plan. At this step, the type of addressing is chosen for networks (autoconf, DHCPv6, static), prefixes are explicitly assigned to some networks or links (DMZ, backbones)...

It is also important at this point to take into account IPv6 specific features that may be deployed on the network, such as IPv6 Mobility... At this point, the routing protocol to be deployed must also be chosen, in order to generate the appropriate configurations. Static routing is an option, but all the commonly used protocols have an IPv6 version and are eligible, such as RIPng [15], OSPFv3 [4] or BGP [16].

4.1.6 Determine an addressing plan

With the site prefix, the constraints and the network topology, define an addressing plan for the site, respecting the prefix aggregation.

4.1.7 Define and set firewall rules

Once the addressing plan is defined, replicate IPv4 security policy to IPv6, while taking into account IPv6 specificities and the new features that may be deployed on the network.

Then, set this new policy in the firewalls, at network level, and locally on the nodes for which it is required.

4.1.8 Update DNS entries

DNS is one of the main issues of the transition. Before enabling IPv6 at network level, the DNS servers must be configured to respond to IPv6 queries. The database should be updated by adding the required AAAA records, but these entries should not be announced at that point, as the addressing has been performed yet.

At this point, for hosts performing autoconfiguration, the name server can be configured to perform Dynamic DNS Updates [22], which can be secured by using the TSIG [21], TKEY [8] or SIG(0) [7] techniques . This implies that private/public keys have been propagated on the network, which may be fastidious if the network is important.

Thus, it is important to identify which nodes should appear in the DNS database, at least in a first step (e.g. servers and core services). After the transition, this database can be updated at any time with the other nodes.

4.1.9 Configure routing infrastructure and address routers

As all firewall rules have been set and the addressing plan determined, it is now possible to configure the routing infrastructure. The first step is to address the routers interfaces and the links end points. Then, assign the prefixes to the networks and configure the routing itself (routes, routing protocols).

In case of usage of VLANs, switches must be configured too at this step. In all cases, if we have network switches with remote configuration, an IPv6 address must be assigned to their management interface (if we want to configure them through IPv6).

Once it is configured, the routing must be tested to validate the configuration, but the site prefix must not be advertised outside the network. Internally, autoconfiguration and DHCPv6 servers are configured, but do not advertise themselves (Router Advertisements (RA) are still disabled).

4.1.10 Address nodes

Activate the sending of RA and DHCP servers, in order to address all nodes.

4.1.11 Verify addressing of core services

Verify that core services like DNS, HTTP... do work, and have an IPv6 address. At the same time, test the services from the network point of view to validate the security policy, and modify it if required before exposing the network to Internet.

4.1.12 Advertise the prefix and DNS

Once everything if configured and validated, advertise the prefix to the neighbors routers, the DNS entries... Make the transitioned network visible to the Internet.

At that point, it is important to inform all our partners (clients, providers...) that we performed the transition, and give them the correspondance between IPv4 and IPv6 addresses of the hosts that are alloed to access their network, or that autorize access to our network. That way, they will be able to update their ACLs accordingly.

4.2 Open Questions

Besides this procedure and the issues that it implies, some open questions remain.

What happens for wireless networks, and especially ones with access points. If they do work at level 3, there may be problems, as many of the commercial access points do not work or are unstable in IPv6. What happens with authentification systems and VPN ?

What are the information we can extract from the IPv4 addressing plan ? Can we use the size of the network mask ? Should we add new constraints for a 1 to 1, 1 to n, n to 1, n to m mapping of the subnets ?

Concerning NAT, it is sometimes used not only to share a public IP, but also for security issues. As NAT does not exist in IPv6, how do we map such a network ? Do we need to add specific firewall rules ? Do we simply map with a public network prefix ?

Chapter 5

Conclusion

In this deliverable, we presented the IPv4 to IPv6 transition, the different mechanisms that exist to do it, and the problems raised.

We presented the different topologies we will consider during this, and a first version of a transition procedure. These topologies and procedure will be the basis of our study. We will study each topology and each step of the procedure, and we aim providing algorithms and their implementation to perform automatically a smooth transition.

Bibliography

[1] R. Callon and D. Haskin. Routing Aspects of IPv6 Transition. RFC 2185 (Informational), September 1997.

[2] B. Carpenter and C. Jung. Transmission of IPv6 over IPv4 Domains without Explicit Tunnels. RFC 2529 (Proposed Standard), March 1999.

[3] B. Carpenter and K. Moore. Connection of IPv6 Domains via IPv4 Clouds. RFC 3056 (Proposed Standard), February 2001.

[4] R. Coltun, D. Ferguson, and J. Moy. OSPF for IPv6. RFC 2740 (Proposed Standard), December 1999.

[5] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460 (Draft Standard), December 1998.

[6] R. Despres. IPv6 Rapid Deployment on IPv4 infrastructures (6rd). draft-despres-6rd-02 (Informational), October 2008.

[7] D. Eastlake 3rd. DNS Request and Transaction Signatures (SIG(0)s). RFC 2931 (Proposed Standard), September 2000.

[8] D. Eastlake 3rd. Secret Key Establishment for DNS (TKEY RR). RFC 2930 (Proposed Standard), September 2000.

[9] R. Gilligan and E. Nordmark. Transition Mechanisms for IPv6 Hosts and Routers. RFC 2893 (Proposed Standard), August 2000.

[10] J. Hagino and K. Yamamoto. An IPv6-to-IPv4 Transport Relay Translator. RFC 3142 (Informational), June 2001.

[11] D. Johnson, C. Perkins, and J. Arkko. Mobility Support in IPv6. RFC 3775 (Proposed Standard), June 2004.

[12] H. Kitamura. A SOCKS-based IPv6/IPv4 Gateway Mechanism. RFC 3089 (Informational), April 2001.

[13] S. Lee, M-K. Shin, Y-J. Kim, E. Nordmark, and A. Durand. Dual Stack Hosts Using "Bump-in-the-API" (BIA). RFC 3338 (Experimental), October 2002.

[14] M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, and L. Jones. SOCKS Protocol Version 5. RFC 1928 (Proposed Standard), March 1996.

[15] G. Malkin and R. Minnear. RIPng for IPv6. RFC 2080 (Proposed Standard), January 1997.

[16] P. Marques and F. Dupont. Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing. RFC 2545 (Proposed Standard), March 1999.

[17] E. Nordmark. Stateless IP/ICMP Translation Algorithm (SIIT). RFC 2765 (Proposed Standard), February 2000.

[18] S. Thomson and T. Narten. IPv6 Stateless Address Autoconfiguration. RFC 2462 (Draft Standard), December 1998.

[19] G. Tsirtsis and P. Srisuresh. Network Address Translation - Protocol Translation (NAT-PT). RFC 2766 (Proposed Standard), February 2000. Updated by RFC 3152.

[20] K. Tsuchiya, H. Higuchi, and Y. Atarashi. Dual Stack Hosts using the "Bump-In-the-Stack" Technique (BIS). RFC 2767 (Informational), February 2000.

[21] P. Vixie, O. Gudmundsson, D. Eastlake 3rd, and B. Wellington. Secret Key Transaction Authentication for DNS (TSIG). RFC 2845 (Proposed Standard), May 2000. Updated by RFC 3645.

[22] P. Vixie, S. Thomson, Y. Rekhter, and J. Bound. Dynamic Updates in the Domain Name System (DNS UPDATE). RFC 2136 (Proposed Standard), April 1997. Updated by RFCs 3007, 4033, 4034, 4035.