CHAPTER 12

# The Migration from IPv4 to IPv6

Migrating from IPv4 to IPv6 in an instant is impossible because of the huge size of the Internet and of the great number of IPv4 users. Moreover, many organizations are becoming more and more dependent on the Internet for their daily work, and they therefore cannot tolerate downtime for the replacement of the IP protocol. As a result, there will not be one special day on which IPv4 will be turned off and IPv6 turned on because the two protocols can coexist without any problems. The migration from IPv4 to IPv6 must be implemented node by node by using autoconfiguration procedures (see Section 6.7) to eliminate the need to configure IPv6 hosts manually. This way, users can immediately benefit from the many advantages of IPv6 while maintaining the possibility of communicating with IPv4 users or peripherals. Consequently, there is no reason to delay updating to IPv6!

We have already seen that some IPv6 characteristics are explicitly designed to simplify the migration. For example, IPv6 addresses can be automatically derived from IPv4 addresses, IPv6 tunnels can be built on IPv4 networks, and at least in the initial phase, all IPv6 nodes will follow the *dual stack* approach; that is, they will support both IPv4 and IPv6 at the same time.

This good level of compatibility between IPv4 and IPv6 may cause some users to think that the migration to IPv6 is useless. In the future, the choice of not migrating to IPv6 will limit the possibility of evolving because it will prevent users from accessing new implementations that, starting from 2000, will concern IPv6 only.

IPv6 has been accurately designed, discussed thoroughly, and tested in the field by the IETF and by many other research institutions. A project called *6-Bone* (described in Section 12.3) was created so that users could acquire experience and test the IPv6 protocol stacks.

The years from 1997 to 2000 will be characterized by the adoption of IPv6 by ISPs and users. During 1997, users could still have problems related to the newness of products, but starting from 1998, IPv6 will be part of mass-produced protocols distributed on routers, on workstations, and on PCs. At that point, organizations will begin to migrate, less or more gradually, to IPv6.

The key goals of the migration are as follow:

- IPv6 and IPv4 hosts must interoperate.
- The use of IPv6 hosts and routers must be distributed over the Internet in a simple and progressive way, with a little interdependence.
- Network administrators and end users must think that the migration is easy to understand and implement.

A set of mechanisms called *SIT (Simple Internet Transition)* has been implemented; it includes protocols and management rules to simplify the migration. The main characteristics of SIT are the following:

- *Possibility of a progressive and nontraumatic transition:* IPv4 hosts and routers can be updated to IPv6, one at a time, without requiring other hosts or routers to be updated simultaneously.
- *Minimum requirements for updating:* The only requirement for updating hosts to IPv6 is the availability of a DNS server to manage IPv6 addresses. No requirements are needed for routers.
- *Addressing simplicity:* When a router or a host is updated to IPv6, it can also continue to use IPv4 addresses.

- *Low initial cost:* No preparatory work is necessary to begin the migration to IPv6.

Mechanisms used by SIT include the following:

- A structure of IPv6 addresses that allows the derivation of IPv6 addresses from IPv4 addresses.
- The availability of the dual stack on hosts and on routers during the transition—that is, the presence of both IPv4 and IPv6 stacks at the same time.
- A technique to encapsulate IPv6 packets inside IPv4 packets (tunneling) to allow IPv6 packets to traverse clouds not yet updated to IPv6.
- An optional technique that consists of translating IPv6 headers into IPv4 headers and vice versa to allow, in an advanced phase of the migration, IPv4-only nodes to communicate with IPv6-only nodes.

The SIT approach guarantees that IPv6 hosts can interoperate with IPv4 hosts initially on the entire Internet. When the migration is completed, this interoperability will be locally guaranteed for a long time. This capability allows for the protection of investments made on IPv4; simple devices that cannot be updated to IPv6—for example, network printers and terminal servers—will continue to operate with IPv4 until they are no longer used.

The possibility of a gradual migration allows manufacturers to integrate IPv6 in routers, operating systems, and network software when they think that implementations are stable and users to begin the migration at a time they consider the most appropriate.

Migration problems are described in RFC 1933[1]. The following sections of this chapter are dedicated to describing these problems.

## 12.1 Tunneling

As we mentioned in the introduction, while the IPv6 routing infrastructure is being deployed, the routing will continue to be based on IPv4. Tunneling techniques (see also Section 7.5.6) allow use of IPv4 networks to carry the IPv6 traffic.

Hosts and routers supporting the dual stack (also called IPv4/IPv6 nodes) can use tunnels to route IPv6 packets over IPv4 regions, as shown

in the example in Figure 12-1.

In this example, host A sends the native IPv6 packet to router R1, which retransmits the packet in an IPv4 tunnel to router R2, which finally transmits it as a native IPv6 packet to host B. In this case, the tunnel is managed by R1 and R2.

From the encapsulation point of view, implementing a tunnel means encapsulating an IPv6 packet inside an IPv4 packet, as shown in Figure 12-2.

In the example shown in Figure 12-2, the IPv6 header will contain addresses A and B, and the IPv4 header will contain addresses R1 and R2.

### 12.1.1 Alternative Tunneling Schemes

During the migration, the tunneling technique can be used in the following ways:

- *Router-to-router:* IPv6/IPv4 routers interconnected by an IPv4 infrastructure can tunnel IPv6 packets between themselves. See Figure 12-3(a).
- *Host-to-router:* IPv6/IPv4 hosts can tunnel IPv6 packets to an intermediary IPv6/IPv4 router that can be reached via an IPv4 infrastructure. See Figure 12-3(b).
- *Host-to-host:* IPv6/IPv4 hosts that are interconnected by an IPv4 infrastructure can tunnel IPv6 packets between themselves. See Figure 12-3(c).

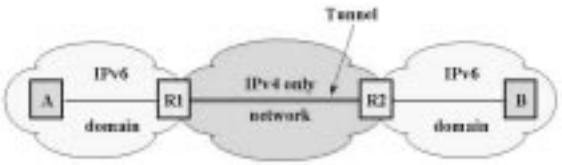



**Figure 12-1**
IPv6 over Ipv4 Tunneling

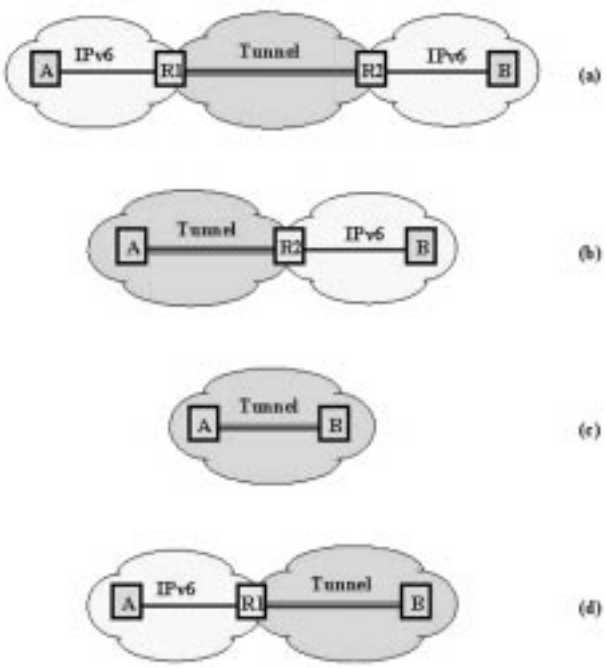

**Figure 12-2**
IPv6 over IPv4 Encapsulation

- *Router-to-host:* IPv6/IPv4 routers can use tunnels to reach an IPv6/IPv4 host via an IPv4 infrastructure. See Figure 12-3(d).

In the first two tunneling methods—router-to-router and host-to-router—the IPv6 packet is tunneled to a router; therefore, the endpoint of this type of tunnel is a router that must decode the IPv6 packet and forward it to its final destination. No relationship exists between the router address and the final destination address. For this reason, the router address that is the tunnel endpoint must be manually configured. This type of tunnel is called a *configured tunnel*.

In the last two tunneling methods—host-to-host and router-to-host—the IPv6/IPv4 packet is tunneled from a host or from a router to its destination host. In this case, the tunnel endpoint address and the destination host address are the same. If the IPv6 address used for the destination node is an IPv4-compatible address (see Section 4.6.8), the tunnel endpoint IPv4 address can be automatically derived from the IPv6 address, and therefore no manual configurations are necessary. These tunnels are also called *automatic tunnels*.

**Figure 12-3**
*Tunneling schemes*



### 12.1.2 IPv6 Addresses with Embedded IPv4 Addresses

IPv6 addresses with embedded IPv4 addresses have the format shown in Figure 12-4, and they have a syntax of the type `::10.1.3.4` (see Section 4.6.8). They must not be confused with IPv4 addresses whose syntax is `10.1.3.4`.

### 12.1.3 MTU

The encapsulating node can also transmit large IPv6 packets (up to 65,535 20-octet packets, because the IPv4 header is 20 octets long) by delegating the fragmentation problem to the IPv4 level. This approach, even if theoretically possible, would be inefficient for the following reasons:

- It would result in more fragmentation than needed. In fact, the loss of an IPv4 fragment would cause the retransmission of the entire IPv6 packet and therefore also of fragments that correctly reached the destination.
- The fragmentation occurring at one endpoint of the tunnel should be removed at the other endpoint. For tunnels that terminate at a router, this process would require additional memory in the router to contain fragments waiting to be reassembled.

Therefore, the fragmentation at tunnel endpoints can be minimized by recording the tunnel's IPv4 Path MTU.

The algorithm used to deal with this problem is described in RFC 1933[1] and reported in Section A.5 of Appendix A.

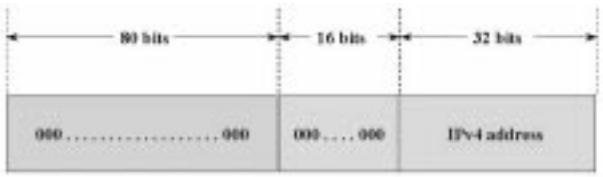



**Figure 12-4**
IPv6 addresses with embedded IPv4 addresses

### 12.1.4 Hop Limit

In IPv6, a tunnel is like a single point-to-point link, and each tunnel corresponds to a hop. The Hop Limit field of the IPv6 header is therefore decremented by one when an IPv6 packet traverses a tunnel, independently from the number of IPv4 links the tunnel consists of.

### 12.1.5 Default Configured Tunnel

An IPv6 node connected to a purely IPv4 network can reach other IPv6 nodes only if a *default configured tunnel* has been defined. It is a tunnel toward an IPv6/IPv4 router that is configured in a way similar to a default route. All the IPv6 traffic will be sent to the IPv6/IPv4 router on the default configured tunnel. This type of tunnel allows testing of IPv6 even on a single host!

## 12.2 Dual Stack Approach

The dual stack approach consists of providing hosts and routers with IPv6 and IPv4 protocol stacks. In the case of an IPv6/IPv4 host, a possible organization of protocol stacks is shown in Figure 12-5.

The dual stack approach doesn't necessarily require the ability to create tunnels, whereas the ability to create tunnels requires the dual stack approach. In general, both approaches are provided by IPv6/IPv4 implementations.

The following is a simple description of the way the dual stack approach operates:

- If the destination address used by the application is an IPv4 address, then the IPv4 protocols stack is used.
- If the destination address used by the application is an IPv6 address with an embedded IPv4 address, then IPv6 is encapsulated inside IPv4.
- If the destination address is an IPv6 address of another type, then IPv6 is used, possibly encapsulated in the default configured tunnel.

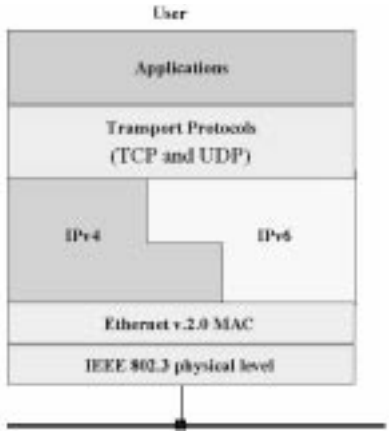



**Figure 12-5**
*The dual stack approach*

As a matter of fact, many more cases can be considered, and a more complete discussion of this topic can be found in a dedicated section of RFC 1933[1], reported in Section A.6 of Appendix A.

Moreover, we must consider that a user normally provides the application with a name, not with an address. This name must be translated into an address by using the DNS (see Section 2.11). In the DNS, only the IPv4 address (record A), only the IPv6 address (record AAAA), or both of them can be stored for each name. In the last case, deciding whether to use the IPv4 address or the IPv6 address is not easy, and the choice is the result of much consideration.

First, determining whether the node has an IPv6 direct connectivity is necessary. If not, the use of the IPv6 address will require the transmission of an IPv6 packet in an IPv4 tunnel. This approach can be less convenient than the use of native IPv4 or even impossible if the node cannot use tunnels.

## 12.3 6-Bone

The 6-Bone project (`http://www-cnr.lbl.gov/6bone/`) is a spontaneous derivation of the IETF IPng working group, and its aim is to implement and test IPv6 protocols with the final goal of replacing IPv4 with IPv6 on the Internet. 6-Bone is an informal collaboration between several research institutions located in Northern America, Europe, and Japan.

A strategic phase of the migration from IPv4 to IPv6 is represented by the implementation of an IPv6 backbone covering the entire Internet and

able to transport IPv6 packets. As in the case of the present Internet IPv4 backbone, the IPv6 backbone will consist of many ISPs and of user networks interconnected to form the new Internet. Until protocols of the IPv6 stack will be widely available and tested, with particular reference to the interoperability of implementations, ISPs and users may not want to migrate production IPv4 routers to avoid risks. Therefore, identifying a way to provide an IPv6 connectivity on the entire Internet without modifying the present IPv4 Internet is necessary in order to test IPv6 protocols and to use them as soon as possible.

***NOTE:*** *6-Bone, which is a virtual network layered on the present IPv4 Internet, provides the routing of IPv6 packets because not all routers currently available can correctly manage the IPv6 routing. The network consists of "islands" providing an IPv6 direct connectivity (usually LANs) interconnected by virtual point-to-point channels (tunnels). Tunnels' endpoints are either single workstations supporting IPv6 or routers supporting IPv6.*

6-Bone is a time-oriented project. In fact, as time goes by and with the growth of the reliability and routing of IPv6 packets on routers, IPv6 will be available by default on new routers and on updated software releases, and 6-Bone will disappear as agreed by its designers. It will be transparently replaced by an IPv6 global connectivity offered by ISPs and by user networks.

The goal of 6-Bone is to provide an environment in which the transport of IPv6 packets can be tested and users are allowed to gain the required experience. It isn't aimed at creating a new and permanent interconnection architecture.

6-Bone is trying to involve as many ISPs and users as it can to spread the experience on IPv6 as much as possible and to create an easy migration to IPv6 itself.

### 12.3.1 The 6-Bone Node at Politecnico di Torino

Figure 12-6 shows the 6-Bone node implemented at Politecnico di Torino, Italy, in September 1997.


**Figure 12-6**
*The 6-Bone node at Politecnico di Torino*

Figure 12-7 shows a dump of the Telebit router in which we can see IPv6 addresses manually configured on the local network and those automatically learned through the RIP protocol. Furthermore, we can estimate the role of tunnels in 6-Bone.

## 12.3.2 Registration to RIPE-NCC

Organizations willing to participate in the 6-Bone experiment should register with the RIPE-NCC. Figure 12-8 shows, as an example, the registration form of Politecnico di Torino.

**Figure 12-7**
*Dump of the Telebit router*

```
% use ip routing 3
% show ip v6route
Route to:                                          Out Interface:
      Metric: Source:
::130.192.26.253/128                                          1.0
      0 IGP Configured
::192.168.0.26/128                                         atm.0
      0 Configured path
3ffe:300::/24                                       cselt.internet
      5 IGP RIPv6
3ffe:301:dec0::/44                                  cselt.internet
      12 IGP RIPv6
3ffe:301:dec1::/48                                  cselt.internet
      12 IGP RIPv6
3ffe:400::/24                                       cselt.internet
      5 IGP RIPv6
3ffe:501:402:a00::/64                               cselt.internet
      8 IGP RIPv6
3ffe:900::/24                                       cselt.internet
      4 IGP RIPv6
3ffe:a00::/24                                       cselt.internet
      12 IGP RIPv6
3ffe:c00::/24                                       cselt.internet
      4 IGP RIPv6
3ffe:c00:0:1::/64                                   cselt.internet
      7 IGP RIPv6
3ffe:f00::/24                                       cselt.internet
      3 IGP RIPv6
3ffe:1000::/24                                      cselt.internet
      3 IGP RIPv6
3ffe:1001:1::/80                                    cselt.internet
      3 IGP RIPv6
3ffe:1001:1:0:0:0:0:1/128                           cselt.internet
      1 IGP Configured
3ffe:1001:1:0:0:0:0:2/128                             cselt.sirius
      1 Configured Peer
3ffe:1011::/32                                           default.1
      1 IGP Static path
3ffe:1011:101:e00::/80                                  default.1
      1 IGP Configured
3ffe:1011:101:e00:0:bd:0:1111/128                      default.1
      0 IGP Configured
3ffe:1011:111:1111:0:1111:1111:1111/128          cselt.internet
      0 IGP Configured
3ffe:1011:111:1111:0:2222:2222:2222/128            cselt.sirius
      0 Configured path
3ffe:1011:111:2222:0:1111:2222:1111/128                    unimi
      0 IGP Configured
3ffe:1011:111:2222:0:1111:2222:2222/128                    unimi
      1 IGP Configured
```

*(Continues)*

**Figure 12-7**
*Continued.*

```
3ffe:1011:111:2222:0:1111:3333:1111/128                unibo
      0 IGP Configured
3ffe:1011:111:2222:0:1111:3333:2222/128                unibo
      1 IGP Configured
3ffe:1011:200::/40                                       unimi
      1 IGP Static path
3ffe:1011:300::/40                                       unibo
      1 IGP Static path
3ffe:1100::/24                                    cselt.internet
      3 IGP RIPv6
3ffe:1200::/24                                    cselt.internet
      5 IGP RIPv6
3ffe:1300::/48                                    cselt.internet
      5 IGP RIPv6
3ffe:1300:1::/48                                  cselt.internet
      5 IGP RIPv6
3ffe:1d00:1::/48                                  cselt.internet
      12 IGP RIPv6
3ffe:1d00:1:100::/64                             cselt.internet
      10 IGP RIPv6
3ffe:1dec::/32                                    cselt.internet
      12 IGP RIPv6
3ffe:2000:0:1:0:0:0:2/127                        cselt.internet
      8 IGP RIPv6
3ffe:2100::/24                                    cselt.internet
      4 IGP RIPv6
5f00::/8

More (Y/N)?n

% show ip v4route
Route to:          Interface:         Metric: Source:
      NextHop:
          0.0.0.0/0        default.2        1 IGP Static
      path
     130.192.0.0/16              1.0       22 IGP Own Domain
    130.192.26.0/24        default.2       20 IGP Configured
 130.192.26.253/32              1.0        0 IGP Configured
     192.168.0.0/24           atm.0       22 Own Domain
    192.168.0.26/32           atm.0        0 Configured
      path
%
```

**Figure 12-8**
*Example of registration form to RIPE-NCC*

```
ipv6-site:   POLITO
origin:      AS5456
descr:       Politecnico di Torino
descr:       Torino, ITALY
location:    45 03 52.2 N 07 39 43.2 E 250m
country:     IT
prefix:      5F15:5000::/32
application: ping girasole-v6.ipv6.polito.it
application: ping telebit-v6.ipv6.polito.it
application: ping ellen-v6.ipv6.polito.it
application: ping alice-v6.ipv6.polito.it
tunnel:      IPv6 in IPv4 telebit.ipv6.polito.it ->
     polo.cefriel.it    CEFRIEL STATIC
tunnel:      IPv6 in IPv4 telebit.ipv6.polito.it -> schu-
     bert.crs4.it   CRS4 STATIC
tunnel:      IPv6 in IPv4 telebit.ipv6.polito.it ->
     telebit.cselt.it   CSELT RIPng
tunnel:      IPv6 in IPv4 telebit.ipv6.polito.it ->
     sun1.spfo.unibo.it UNIBO STATIC
tunnel:      IPv6 in IPv4 telebit.ipv6.polito.it ->
     phoebe-v6.ip6.dsi.unimi.it UNIMI STATIC
contact:     SG389-RIPE
remarks:     OpenBSD/NRL, Sun Solaris, DEC RouteAbout Ac-
     cess EW/IPv6, Telebit
remarks:     Running Bind 4.9.5 on ns.ipv6.polito.it
remarks:     our modified NRL distribution is available at
     ftp.ipv6.polito.it
remarks:     ipv6-site is operational since 11/1996
url:         http://www.ipv6.polito.it
notify:      silvano.gai@polito.it
changed:     spera@csp.it 19970324
changed:     auto-dbm@ISI.EDU 19970331
changed:     rivetti@csp.it 19970609
changed:     spera@alp.net 19970917
source:      6BONE

% Rights restricted by copyright. See
     http://www.ripe.net/db/dbcopyright.html

person:      Silvano Gai
address:     Dip. Automatica e Informatica
address:     Politecnico di Torino
address:     Corso Duca degli Abruzzi 24
address:     I-10129 Torino
address:     Italy
phone:       +39 11 5647013
```

## REFERENCES

[1]R. Gilligan, E. Nordmar, *RFC 1933: Transition Mechanisms for IPv6 Hosts and Routers*, April 1996.

[2]R. Hinden, J. Postel, *RFC 1897: IPv6 Testing Address Allocation*, January 1996.