INTERNET-DRAFT
NGTRANS Tools Working Group
December 18,1998
expires in 6 month

Laurent Toutain
Hossam Afifi
ENST Bretagne

Dynamic Tunneling: A new method for the IPv4-IPv6 Transition

<draft-ietf-ngtrans-dti-00.txt>

Status of this Memo

Abstract

We propose to use an IPv6 tunneling mechanism and encapsulate IPv4 packets into IPv6 packets. This eases the transition and routers do not need to support IPv6 and IPv4 routing tables. The IN_ADDR.arpa domain is used to find IPv4 to IPv6 correspondence in a way that does not prevent the asymmetrical routing and multi-homing.

The model also simplifies IPv4 address management since the address has no more a localization function but only an identification function.

1. Introduction

The co-existence of IPv6 with IPv4 will be a temporary phase before a progressive transition to the IPv6 protocol even if no limit has been set for the transition period. The first phase in the transition was to install new IPv6 subnetworks connected by tunneling justified by the relatively small number of IPv6 networks compared to the global internet.

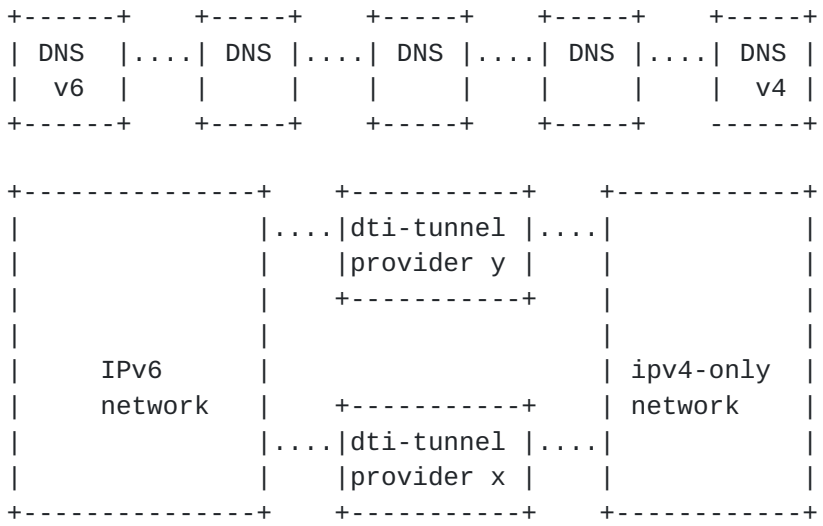
The goal of this work is to propose a simple and flexible

interconnection mechanism where IPv6 and IPv4 networks can be separated, but where IPv6 and IPv4 applications can transparently exchange information. This gives the opportunity to use IPv6 in combination with other protocols for a more flexible routing.

The next section introduces native IPv6 networks and hosts and places the study in its context. First we describe the model we choose to study. We analyze current transitions propositions. In a second part we describe the proposal. We cover the cases where connections are initiated from the IPv6 and from IPv4 sides respectively.

2. Transition Model

In our view a general model describing the transition period is depicted in figure 1. We consider a domain running mostly IPv6 but where some IPv4 equipments or applications still remain used. This domain is connected to an IPv6-only provider. Somewhere on the network a third party provider is able to link the IPv4 and IPv6 domains. We solve the case where IPv6 equipment must talk with equipment in IPv4-only domains and vice versa. We suppose that no change can be made to the IPv4-only equipment and applications. The transition mechanism must be as transparent as possible to keep the advantages of the IPv6 autoconfigation.



A DNS is running on each domain. Queries to the DNS are done in the respective native protocol (we suppose that they are interconnected but the way of interconnection is beyond the scope of this draft).

2.1 The double stack

[RFC 1933] defines a transition mechanism based on a double stack. A host will either send packets in IPv4 or IPv6 depending on the protocol used by the destination. The IPv4 mapped addresses allow applications compiled with IPv6 system calls to talk with IPv4 only applications.

This method is currently used in the early phase of the transition. The main drawbacks are the following. An IPv4 address must be allocated to each equipment. If we map this onto our transition model previously described, it means that the IPv6 domain is included in the IPv4 world. Routers must be configured for the two protocols and IPv4 applications must be slightly modified and recompiled to be adapted to the IPv6 API.

2.2 The SIIT proposal

This proposal is similar to Network Address Translation (explained later), used in IPv4 networks to exchange a private IPv4 address to a public IPv4 address. It is difficult with the SIIT proposal to deal with applications sending addresses (such as FTP or RTP flows). In that case, some Application Level Gateways acting as proxy will be required. The proposal focuses on separate IPv6 and IPv4 domains. If in the IPv6 domain some equipment wants to talk with an IPv4 domain, it will use an automatic IPv4 address allocation of the double stack. In that case, routers will have to administrate IPv4 and IPv6 routing tables.

Like NAT (described hereafter), this is a one way translation. It means that outside IPv6 domains, IPv4 applications cannot initiate communications with IPv6 hosts. The context for header translation in the SIIT box can be established only when an exiting IPv6 packet leaves the domain.

2.3 The NATPT Proposal

Unlike SIIT [SIIT], NAT-PT is designed to provide transparently end-to-end solutions. A pool of IPv4 addresses is used to be assigned to IPv6 hosts dynamically in response to any request for packets leaving one of the boundaries. These assigned addresses in turn are used to transparently replace the original addresses used by IPv6 end nodes and vice versa. This proposal allows translation in both ways since the context inside the transition box is this time established by the DNS.

NATPT does not solve the problem of applications sending IP addresses in the payload. DNS and NATPT must be combined to allow the establishment of the context. This is generally not the case if the translation is not directly made by the IPv6 domain. The DNS request

may follow another path that does not go through the third party provider assuring the transition.

2.4 The AIIH Proposal

The AIIH proposal aims at using a combination of DHCPv6 and the DNS to establish a transition between IPv6 and v4 in both directions. This proposal is complementary to the NAT -PT and SIIT since it focuses on topology where IPv4 and IPv6 are in the same domains.

For an IPv6 host to participate in the AIIH mechanism, it MUST have a dual IP layer, supporting both an IPv4 and IPv6 stack. When an IPv6 host wants to talk with an IPv4 one, the DNS will not answer with an IPv6 record but with an IPv4 one. The IPv6 will request a temporary IPv4 address. Mechanisms such as DHCP are currently available to do such assignment. The opposite case, where an IPv4 host wants to talk with a IPv6 host is more difficult to solve since no protocol is currently available to do a automatic assignation.

In our view, there are two main drawbacks in this proposal. The router must be configured both for IPv4 and IPv6 protocol and the assignment of IPv4 addresses is difficult since the network topology must be taken into account.

3. The Dynamic Tunneling Interface

In the IPv6 case networks are largely spread, we propose to reverse the tunneling and encapsulate IPv4 packet into IPv6 packets. This would ease the transition since routers need only to be configured with IPv6 routing tables. The case of dynamic attribution of IPv4 addresses will also be simplified since the address has no longer a localization function but only an identification function. The system has just to maintain uniqueness of the address allocation.

We think this method can be used either in domains where IPv4 applications cannot be recompiled, but where the system stack can be changed to take benefit from an IPv6 network or with IPv4 only legacy equipment. We will explain later scenarios where an IPv4 only domain wants to join an IPv6 equipment.

3.1 IPv6 to IPv4

We start describing the general procedure for an IPv6 host to communicate with an IPv4 only application. From our assumptions we suppose that the IPv6 equipment MUST support a dual stack.

When an equipment wants to join an IPv4 host, it may directly use an

IPv4 address or resolve a domain name from the DNS and obtain the correspondent IPv4 destination address.

We propose to intercept the packet in a new interface that we call dti for dynamic tunneling interface. It may be intercepted by the host through the IPv4 routing table lookup or through dynamic network libraries hookups but this is implementation dependent. We try hence to offer transparently the dti functionalities with no changes in the applications. The main difference with AIIH is that IPv4 packets will not be directly sent over the network but first filtered by the dti.

The dti interface systematically encapsulates the IPv4 packet into an IPv6 packet. It needs to find the destination IPv6 address and also needs to obtain for itself an IPv4 address. We propose to use the DNS to make this resolution. The procedure is explained later. Once a result is obtained, it should be cached after the first resolution. There are many possible results for this IPv6 resolution.

- The destination has an IPv6 address.

- It has only an IPv4 address.

In the first case, the dti returns an IPv6 destination address. It is possible to establish an end-to-end tunnel and there is no need to obtain a global IPv4 address. If the dti has not yet a local address it will configure the IPv4 stack and use the address in the encapsulation. The local IPv4 address should be de-allocated after a certain idle time.

In case of an IPv4-only destination equipment, we propose to implement tunneling boxes inside the network that we call dti-tunnels. These boxes should be as close as possible to IPv4 resources to reduce routing of native IPv4 packets. They can also be located on a third party provider network if no possible internetworking between IPv6 and IPv4 can be achieved in the local domain. The dti must obtain a dti-tunnel IPv6 address. This may be statically configured in each dual stack host. It could be obtained as mentioned in AIIH proposal by means of a RR record in the destination domain name (we propose to add in the DNS a new entry for NGTRANS-TUNNELING) or via more advanced methods like NHRP.

The dti sends a DHCP(v4) query to the dti-tunnel encapsulated in IPv6. The dti-tunnel assigns an IPv4 address to the dti. In the general case, the assignment can be static through an IPv4 pool of addresses in the dti-tunnel. The dti-tunnel can act as forwarder and obtain dynamically an IPv4 address through a conventional DHCP(v4) procedure.

The dti is now able to send the packets to the dti-tunnel that decapsulates the IPv6 header and sends them to the destination through IPv4 network. The procedure is hence quite similar to NAT but does not need any address translation at boundaries.

3.2 DNS Resolution in the dti

The dti queries the DNS when a packet is received from an IPv4 application to the dual stack. A query to the IN_ADDR.arpa domain is sent to obtain the original destination domain name. This is the main contribution in the dti tunneling model. Once this is achieved a second query is sent to the DNS for the resolution of an AAAA type address. If an answer is obtained a direct tunnel can be established. Otherwise, the dti may, as it is mentioned in the AIIH proposal look for a RR record with the AAAA type to be used as the dti-tunnel. The dti proceeds as explained before.

During the name resolution the dti should check whether the destination is located on the local intranet or not. If it is in the same domain as the origin then the dti should use a local dti-tunnel. We illustrate the DNS procedure by an example: the dti receives a packet to be sent over IPv4 to the destination 192.44.77.131. It will make a query to the DNS asking for qtype=ptr. The query itself will look like:

131.77.44.192.in-addr.arpa.

and the answer :

131.77.44.192.in-addr.arpa name = abc.xyz.com.

This entry describing the host abc will give more information on its addresses and tell whether there is an IPv6 address for the host. The final answer if there is an IPv6 address will be:

```
abc                             IN      A       192.44.77.131
                                IN      AAAA
3ffe:0305:1002:0001:0a00:2bff:fe1c:b0df
```

4. IPv4 to IPv6

We present the second part of the dti model where an IPv4 only equipment needs to reach an IPv6 only one.

If the IPv4 host has already an IPv4 global address to the destination it will send the packet through the normal routing procedures. Once the packet arrives to any dti-tunnel equipment the same DNS resolution as in the dti case (section 3.2) should be done.

We try to find an IPv6 address for the IPv4 destination by consulting the IN_ADDR.arpa domain and once it is obtained we encapsulate the packet in IPv6 and send it to the destination.

If the IPv4 only host sends a DNS query to resolve the destination domain name into an IPv4 address the query may succeed and nothing more is needed from the dti model. The packet will be sent using IPv4 and will arrive to the dti-tunnel as previously described. If however the resolver fails to locate an IPv4 address for the requested domain name we use the dti model to execute an additional procedure instead of returning the name-error to the IPv4 application. The dti will make a second query for the same domain name asking this time for an IPv6 address resolution. If there is no IPv6 available then the resolver will return a name error.

If however an IPv6 address is found for the requested domain name the resolver library should trigger a special dti procedure for the tunneling. This is explained in the next paragraph.

IPv4 to IPv6 tunneling

The resolver must have a pre-configured dti-tunnel IPv4 address for its domain. It should send a query to this equipment asking to assign an IPv4 address to the IPv6 destination. The dti-tunnel obtains an IPv4 address from a local pool or using DHCPv4. It remotely configures the IPv4 destination stack and updates the in_addr.arpa domain name with a pointer entry to the IPv6 host. Since the destination may be multi-homed, it is also necessary to update the destination DNS database with the new IPv4 address. We present hereafter a solution to avoid the need to update the destination domain name. Note also that all these procedure will use time to live fields that will protect the system from any malicious attack trying to consume all IPv4 available addresses in a site. The following example traces the steps in the IPv4/IPv6 direction.

The resolver intercepts a query to lmn.xyz.com asking for an IPv4 address (A type query). The query fails and the resolver asks again for an IPv6 address. This time the query succeeds and an IPv6 address is returned. The resolver sends a special query to its pre-configured dti-tunnel asking for the assignment of an IPv4 address to the lmn.xyz.com destination. The dti-tunnel makes a DHCPv4 query and obtains the required address (192.108.119.33 for example). It should use a special procedure to configure (lmn.xyz.com) with the new address and update its own DNS:

33.119.108.192.in-addr.arpa name = lmn.xyz.com.

In case the dti-tunnel uses an additional DHCP(v4) feature that

enables to remotely configure the destination dual stack with the newly assigned IPv4 address, it returns the IPv4 address to the resolver and stops. If however a different procedure is used to configure the remote destination host with the IPv4 address another DNS update in the destination domain will also be needed:

```
lmn                     IN      A 192.119.108.33
                                AAAA
3ffe:0305:1002:0001:0a00:2bff:fe1c:b099
```

Any dti-tunnel that will receive the packet will be able to find the IPv6 address and establish the tunnel by consulting the destination domain name. Again if the use of a DHCPv4 server driven procedure is adopted then when a different dti-tunnel receives an IPv4 assignment request for the IPv6 host it will try to configure remotely the destination stack through DHCPv4. If an address is already assigned for that host then the DHCPv4 configuration will fail and the current IPv4 destination address will be returned in the error message (see section 5). The dti-tunnel will return the address to the resolver and the communication will start.

The usage of a new server driven DHCPv4 request will improve the security environment since the dti-tunnel will not need to update the destination domain name.

4.1 Path MTU

When packets are sent to the dti-tunnel the header length may change the link MTU. This can be either detected by the PMTU detection or if necessary by means of the IPv6 fragmentation extension.

5. Required Additional Features

The dti model will require some additional protocol features. Note that most of them are also required in the AIIH proposal.

In the reverse direction (from IPv4 to IPv6) two new procedures are required. The dti resolver needs to send a query to the dti-tunnel asking for the assignment of an IPv4 address. The dti-tunnel has to remotely configure the destination IPv4 stack with a new address. This whole procedure could be done by the DHCPv4 protocol if an additional feature enables the DHCPv4 server to initiate an IPv4 assignment procedure without a previous solicitation from the client.

An additional entry in the DNS for ipv4/ipv6 capability tunnel would also be very useful (for all the proposals) in order to find for each domain the closest tunneling equipment.

6. Bibliography

[AIIH] J. Bound. Assignment of IPv4 Global Addresses to IPv6 Hosts (AIIH). <draft-ietf-ngtrans-assgn-ipv4-addrs-00.txt> Expired.

[SIIT] E. Nordmark,Stateless IP/ICMP Translator (SIIT), <draft-ietf-ngtrans-siit-02.txt>, Work in progress, November 1998.

[NAT] G. Tsirtsis, P. Srishuresh. Network Address Translation - Protocol Translation (NAT-PT). <draft-ietf-ngtrans-natpt-03.txt>.

7. Authors Addresses

Laurent Toutain Hossam
Afifi ENST
Bretagne 2 rue de la chataigneraie BP 78
35512 Cesson Sevigne {toutain,
afifi}@rennes.enst-bretagne.fr Tel: (+33) 2 99 12 70 20 -
Fax: (+33) 2 99 12 70 30