Internet Engineering Task Force
Internet Draft
Document: draft-palet-v6ops-proto41-nat-03.txt

Category:

Expires: April 2004

Jordi Palet
Cesar Olvera
Consulintel
David Fernandez
UPM

October 2003

# Forwarding Protocol 41 in NAT Boxes

draft-palet-v6ops-proto41-nat-03.txt

## Status of this Memo

Abstract

Some IPv4-only NAT boxes/routers allow the establishment of IPv6 tunnels from systems in the private LAN (using private IPv4 addresses) to routers or tunnel servers in the public Internet.

As far as we know [2] this is not a common way of using IPv6 tunnels; the usual way is to finish the tunnel directly in a device with an IPv4 public address.

This behavior provides a big opportunity to rapidly deploy a huge number of IPv6 nodes and networks, without the need of new transition mechanism. This option is very important to facilitate the IPv6 deployment when is not possible to offer native IPv6 or 6to4 [3].

From this point of view, this mechanism should be considered only as a temporary solution until native IPv6 routers, or those that support 6to4, will become widely available.

Not all the IPv4-only NAT boxes/routers support this mechanism, but this document describes this behavior and consequently provides hints that should be applied in the IPv4-only NAT boxes and tunnel brokers to facilitate it.

## Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [4].

Table of Contents



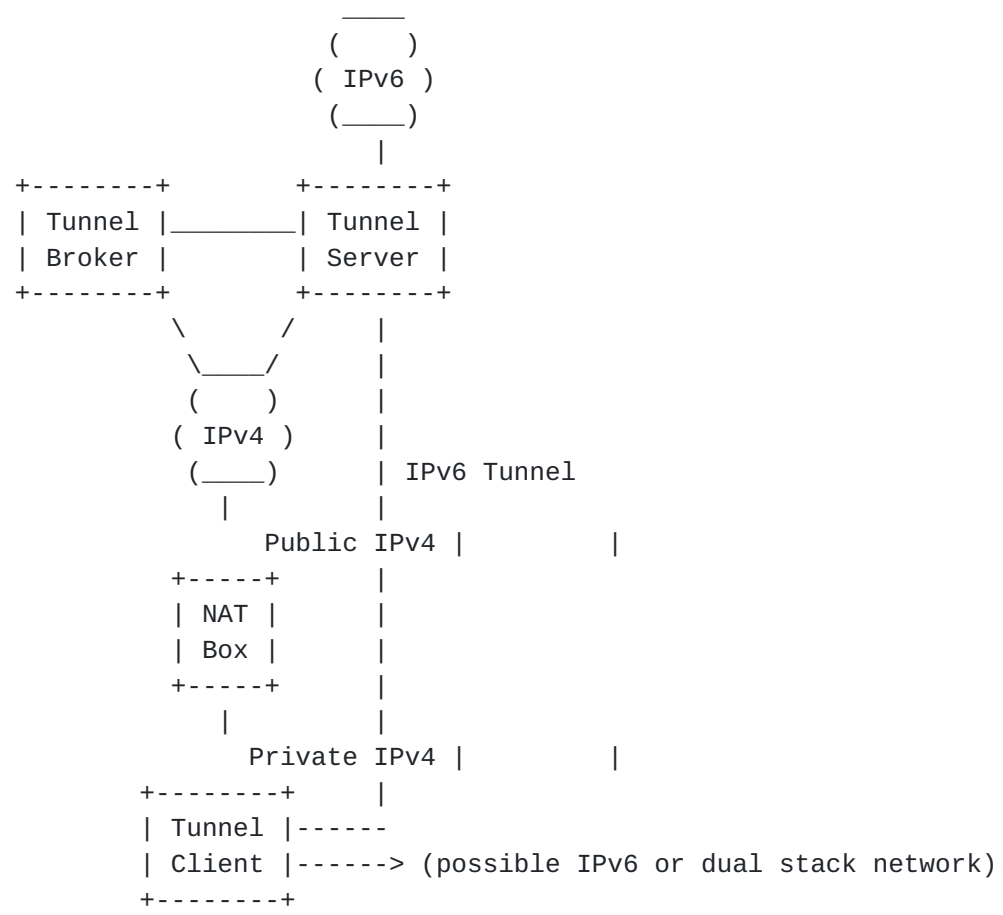
## 1. Introduction

Most of the existing solutions for the transition to IPv6 rely in tunnels assuming that the client end-point is an IPv6 capable router. However, nowadays the installed base of IPv4-only NAT boxes/routers is still quite big, while most of the client operating systems already support IPv6.

The ability of some IPv4-only NAT boxes/routers to establish IPv6 tunnels from systems inside the private LAN (even using private IPv4 addresses) to routers or tunnel servers in the public Internet has been used for some time. However, it has not been documented so far.

The goal of this document is to describe in detail that functionality and to show the rationale behind it, as well as to provide some recommendations for IPv4-only NAT boxes and tunnel broker implementers in order to facilitate its use and deployment.

The basic scenario of the mechanism presented is shown in the Figure below. As can be seen, a Tunnel Client (a host or a router), which is connected to Internet through an IPv4-only NAT box using a private IPv4 address, establishes an IPv6 tunnel to a Tunnel Server with the help of a Tunnel Broker. The mechanism can also be used without a tunnel broker, ending the tunnel in an IPv6 router, which is configured manually.




Typically, IPv6 routers on the Tunnel Server side support the establishment of these tunnels without any additional configuration. However, in the case of some clients under certain operating systems, the tunnel configuration process or the tunnel broker scripts have to be modified to reflect the private/public addressing conversion.

This fact should be taken into consideration by tunnel broker

implementations in future versions, in order to properly create the script in case the client is located in a private network.

This document describes the reasons why this scenario works as it is using present NAT implementations. We consider that exploring this option is very important to facilitate the IPv6 deployment, as it can be used as a temporary fallback solution when neither native IPv6 nor 6to4 mechanisms are available.

The document does not discuss how the local private network is organized, for example, in case the Tunnel Client is an IPv6 router providing IPv6 connectivity to other systems. The behavior in this case should be the same as any other IPv6 native network (that is using stateless or stateful autoconfiguration, or any other typical functionalities like Home Agent, etc).

Although this mechanism is not usable on all existing IPv4-only NAT boxes/routers, the large number of them that already support it gives an opportunity to rapidly deploy a huge number of IPv6 nodes and networks (in case the node behind the NAT is an IPv6 router) without the need of using or designing new transition mechanisms.

The scenario presented has been tested with several IPv4-only NAT boxes that have successfully established IPv6 tunnels between tunnel clients in a private network and tunnel servers in the public Internet. In these test, we have used three well-known Tunnel Broker implementations (BT, Freenet6 and TILAB) as well as manually configured tunnels with routers from several manufacturers.

## 2. Rationale of proto-41 forwarding

As described in RFC 2663 [5]:

"Address translations performed by NAT are session based and would include translation of incoming as well as outgoing packets belonging to that session ... a session is defined as the set of traffic that is managed as a unit for translation. TCP/UDP sessions are uniquely identified by the tuple of (source IP address, source TCP/UDP port, target IP address, target TCP/UDP port). ICMP query sessions are identified by the tuple of (source IP address, ICMP query ID, target IP address). All other sessions are characterized by the tuple of (source IP address, target IP address, IP protocol)."

Basically, what the NAT router does in the scenario presented in this document is a network address translation for protocol identifier 41 (the one used for IPv6 over IPv4 tunnels). The router considers each tuple of the form [source IP address, target IP address, IP protocol (41)] a different session, and typically creates a new proto 41 entry in its table whenever an IPv6 over IPv4 packet flows from the private network to the Internet (as it does, for example, for TCP connections).

## 3. Behavior of different NAT types

As mentioned before, some NAT routers do not support protocol 41 forwarding. They are usually limited to do network address translation for common protocols like TCP, UDP and ICMP. This type of NAT routers will not be considered in this document, although some recommendations for them will be given in section 5.

RFC 2663 [5] distinguishes several types of NAT routers. This document focuses on how proto 41 forwarding works over the two most common types: Traditional NAT and NAPT.

### 3.1 3.1 Traditional (or) Outbound NAT

In traditional NAT routers sessions are unidirectional. This means that, as IPv6 tunnels are treated as any other NAT dynamic session, the tunnel entries are only added to the table whenever an IPv6 packet is sent from the private network to the public Internet, but not with packets flowing in the opposite direction (i.e., coming from the external tunnel endpoint).

Usually, an inactivity timer is started when the NAT entry is created, so that the session (and consequently the tunnel) is deleted if no packets are sent during the inactivity period (a few minutes typically). In case the tunnel entry is deleted due to inactivity, it will be created again whenever a new packet is sent from the private network.

RFC 2663 distinguish between two types of traditional NAT routers: Basic NAT and NAPT. Basic NATs do the address translation by means of a one-to-one association between private and public addresses, so entries on the NAT table have the form [private address, public address].

In the case of NAPT, which is the most widely used at present, each public address can be shared among several private systems, by using different transport ports for each one. Entries in NAT table have the form [source IP address, source TCP/UDP port, target IP address, target TCP/UDP port]. Both types can be combined in the same NAT router.

Support for protocol 41 forwarding in Traditional NAT routers basically means that they should be prepared to manage sessions of the form [source IP address, destination IP address, protocol ID] for protocol 41.

### 3.2 Bi-directional (or) Two-Way NAT

As stated in [5], with a Bidirectional NAT, sessions can be initiated from hosts in the public network as well as the private network. Private network addresses are bound to globally unique addresses, statically or dynamically as connections are established in either direction .

RFC 2663 mentions the use of a DNS-ALG algorithm in order to allow public hosts to communicate with private ones. Basically, whenever a DNS query is made for a private host name, the DNS-ALG in conjunction with Bi-directional NAT answers the query using an available public address and sets the corresponding NAT table entry.

However, this mechanism does not fit proto-41 forwarding requirements, as no names are normally involved when setting up tunnels.

What it is needed in our case is just the basic mechanism included in bi-directional NAT to statically associate one of the private addresses with a public address, only for protocol 41.

In this way, all ingoing IPv6 over IPv4 traffic will be forwarded to the designated internal system and the tunnel will work in a complete bidirectional way, even when no outgoing traffic is generated. IN that case, the inactivity timer will probably not be needed, as the entry on NAT table for outgoing traffic could also be statically configured (by means of a configuration file, http interface, CLI, etc.).

## 4. Applicability

In the case of Basic NAT and NAPT, IPv6 tunnels can only be initiated by inside-to-outside sessions. So in this case, outside-to-inside sessions only work whenever a previous inside-to-outside session has created the proto-41 entry in the NAT table and the inactivity timeout has not been reached.

This fact is only a problem when IPv6 servers or services inside the private network are needed to be accessible from outside. If the traffic is client initiated, the session will be created normally as soon as the first packet is sent, allowing IPv6 communication.

The only way to maintain the session permanently is to constantly send traffic, for example, with a periodic ping from the Tunnel Client, a router solicitation message, or other means. Alternatively, some simple keep-alive protocol could be integrated inside tunnel broker implementations in order to maintain the tunnel.

In the case of Bi-directional NATs, there are means to support also incoming sessions, even when no outgoing traffic haa been generated. However, they require some type of pre-configuration in the IPv4-only NAT box.

To facilitate that, a default configuration could be defined. For example, in the case of simple NAT routers used in most SOHO accesses, the default configuration could include a pre-defined private network address for the LAN interface and a pre-defined private address for the host where all the proto-41 traffic is forwarded.

In summary, the application of proto-41 forwarding procedure allows in both cases the operation of private IPv6 networks connected by means of non-IPv6 aware NAT boxes to tunnel brokers or manually configured tunnels.

The most usual scope of application of the proto-41 forwarding procedure described in this document seems to be SOHO and home environments, but it is not only limited to those scenarios.

## 5. NAT design considerations and recommendations

This document has been written following a survey with users/vendors of different IPv4-only NAT boxes, and the conclusion is that most of the manufacturers support protocol-41 forwarding (78% in our survey). Nevertheless not all support a bidirectional mode (over 22% of the surveyed models do not support it).

NAT boxes should tend to support native IPv6. If this is not feasible, 6to4 should be the second option, and as a last resort, proto-41 forwarding.

6to4 and Proto-41 forwarding can coexist in the same NAT box. In that case, an IPv6 over IPv4 packet received, will be forwarded to the private LAN only if the IPv6 destination does not belong to the local 6to4 /48 prefix. Otherwise it will be decapsulated in the NAT box, following 6to4 procedures. This fact avoids the problems created by mobile users when they visit a network that uses 6to4, in the case they have some automatic proto-41 setup.

New firmware/software versions of the NAT implementations should ensure the support of protocol-41 forwarding, as a temporary solution, while they are not supporting native IPv6 or 6to4.

Proto-41 make sense only in IPv4-only routers, but nevertheless, when these routers are upgraded to support, for example, 6to4, for

compatibility reasons (with existing network configurations), it could be still considered to maintain the support proto-41.

In addition, considering that the code changes needed to support a full bidirectional NAT will be minimum, this option should also be considered, at least as a configurable option, in an easy way by the user (very simple http interface).

Proto-41 adds an inexpensive feature to existing IPv4-only NAT boxes, facilitating the gradual transition to IPv6, while preserving the users investment in the existing IPv4 network.

## 6. Tunnel broker design considerations

New releases of tunnel brokers should provide means to cope with the scenario defined in this document. They should automatically detect it or, at least, they should allow the user to specify manually that a NAT router is present between the tunnel client and server.

According to that the tunnel broker must properly create the script or configuration file that will setup the client tunnel endpoint. In that case they should have requested the public addresses (can be automatically detected) and the local interface ID or name of the tunnel client.

## 7. Security Considerations

It is important to note that IPv6 applications sending traffic over the tunnels described here do not suffer the restrictions that apply to NAT traversal scenarios, because NAT is made to IPv4 packets that transport IPv6 ones, not to IPv6 packets.

Besides, the protection derived from the unidirectional nature of NAT disappears for IPv6. Therefore, some security mechanism (network or personal firewalls) could be necessary to protect IPv6 systems in the private network.

A possible security problem is the one related to the DoS Attack than can be created if a host in the local network, behind the NAT sends IPv6 packets (using protocol 41) to the tunnel endpoint, simulating to be the original "owner" of the tunnel. The behavior of the IPv4-only NAT box will define the success or failure of this attack. In any case, it seems not reasonable that this happens in small networks (SOHO and home environments), where the attacker can be easily identified.

This considerations are generic to transition mechanisms, as described in [6].

## 8. References

1 S. Bradner, "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

2 J. Palet, C. Olvera, D. Fernandez, "IPv6 Tunnels through Routers with NAT", Euro6IX Project, http://www.euro6ix.org/documentation/euro6ix_co_upm-consulintel_wp4_ipv6_tunnels_nat_v1_6.pdf, April 2003.

3 B. Carpenter, K. Moore, Connection of IPv6 Domains via IPv4 Clouds , RFC 3056, February 2001.

4 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

5 P. Srisuresh. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, August 1999.

6 P. Savola, IPv6 Transition/Co-existence Security Considerations , draft-savola-v6ops-security-overview-00, June 2003 (work in progress).

## Acknowledgments

The authors would also like to acknowledge the inputs from Tim Chown, Miguel Angel Diaz, Alain Durand, Jun-ichiro "itojun" Hagino, Keith Moore, Mariana Nikolova, Rute C. Sofia and the European Commission support in the co-funding of the Euro6IX project, where this work is being developed.

## Authors' Addresses

Jordi Palet Martinez
Consulintel
San Jose Artesano, 1
28108 - Alcobendas (Madrid - Spain)
Phone: +34 91 151 81 99
Fax: +34 91 151 81 98
Email: jordi.palet@consulintel.es

Cesar Olvera Morales

Consulintel
San Jose Artesano, 1
28108 - Alcobendas (Madrid - Spain)
Phone: +34 91 151 81 99
Fax:  +34 91 151 81 98
Email: cesar.olvera@consulintel.es

David Fernandez
Technical University of Madrid (UPM)
Ciudad Universitaria s/n
28040  Madrid (Spain)
Phone: +34 91 549 57 00
Fax:  +34 91 336 73 33
Email: david@dit.upm.es

## Intellectual Property Statement



## Full Copyright Statement

## Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.