INTERNET DRAFT
<draft-huitema-v6ops-unmaneval-00.txt>
February 14, 2003
Expires August 14, 2003

C. Huitema
Microsoft
R. Austein
Bourgeois Dilettante
S. Satapati
Cisco Systems, Inc.
Ronald van der Pol
NLnet Labs

Evaluation of Transition Mechanisms for Unmanaged Networks

Status of this memo

Abstract

In a companion paper we defined the "unmanaged networks" scope, which typically correspond to home networks or small office networks, and the requirements for IPv6 transition mechanism in that scope. We start from this analysis and evaluate here the suitability of mechanisms defined in the NGTRANS working group.

1 Introduction

In a companion paper [UNMANREQ] we defined the "unmanaged networks" scope, which typically correspond to home networks or small office networks, and the requirements for IPv6 transition mechanism in that scope. We start from this analysis and evaluate here the suitability of mechanisms defined in the NGTRANS working group.

The requirements for unmanaged networks are expressed by analyzing four classes of application: local, client, peer to peer, and servers, and considering four cases of deployment. These are:

A) Gateway does not provide IPv6

B) ISP and gateway are dual stack
C) Gateway is IPv6 capable, dual stack, ISP is not
D) ISP is IPv6 only

This document analyses the issues involved in the transition from IPv4 to IPv6.

During the transition phase from IPv4 to IPv6 there will be IPv4 only, dual stack or IPv6 only nodes. In this document, we make the hypothesis that the IPv6 only nodes do not need to communicate with IPv4 only nodes; devices that want to communicate with both IPv4 and IPv6 nodes are expected to implement both IPv4 and IPv6, i.e. be dual stack. This issue is discussed in detail in a companion paper [NATPTISSUES].

The issues involved are described in the next sections. This analysis outlines two types of requirement: connectivity requirements, i.e. how to ensure that nodes can exchange IP packets, and naming requirements, i.e. how to ensure that nodes can resolve each-other's names. We also consider the specific security issues that arise during the transition.

2 Meeting case A requirements

In case A, IPv6 capable hosts located behind a NAT need to acquire IPv6 connectivity. In this section, we first evaluate how mechanisms already defined or being worked on in the IETF meet this requirement. We then consider the "remaining holes" and recommend specific developments.

2.1 Evaluation of connectivity mechanisms

In case A, IPv6 capable hosts seek IPv6 connectivity in order to participate with applications in the global IPv6 Internet. The connectivity requirement can be met in two ways: Teredo [TEREDO], or UDP tunnels.

2.1.1 TEREDO

TEREDO is a mechanism designed to provide IPv6 connectivity to hosts behind NATs. Hosts use servers to find out a "mapped" IPv4 address and UDP port; they build an IPv6 address that includes the IPv4 address of their preferred server, and their own mapped IPv4 address and mapped port. A mechanism of bubbles, relayed by the servers, is used for establishing contacts between Teredo nodes, or for discovering the appropriate Teredo relay serving an IPv6 peer; the actual IPv6 packets are carried in UDP packets exchanged directly between the nodes, or exchanged through the relay serving an IPv6 peer.

2.1.2 Simple UDP tunnel

An alternative to TEREDO is to simply establish a tunnel to a "tunnel broker" outside the unmanaged network; in order to traverse the NAT, the IPv6 packets would be carried over UDP. This solution is mentioned as a possible alternative to the bubble mechanism in the TEREDO specification.

2.1.3 Comparison of TEREDO and tunnel solution

The TEREDO and tunnel solutions differ in terms of complexity and operation.

A first difference is the cost of operating the server. TEREDO is designed to minimize the amount of traffic flowing through the server, which only processes small bubbles and never relays traffic; on the other hand, a tunnel server will relay all the packets going to and from the Teredo host. This implies a very different amount of traffic.

To gauge the difference, we consider the case of a host engaging in Voice over IP: it will maintain its address reachable all the time, and it will send a large amount of traffic whenever it is engaged in a conversation. According to classic figures collected by AT&T, the average duration of a conversation is around 100 seconds, and a business telephone is likely to be engaged in a conversation about 10% of the time, which implies starting a conversation on average every 1000 seconds. The average load sent by a tunnel client to the tunnel server will be 10% of the average data rate of the client; assuming a 16kbps codec and 50 packets per second, the data rate of the client sums up to about 51 kbps, hence an average load on the tunnel server of 5 kbps. The load sent by the tunnel client to the tunnel server will be about one exchange of bubble per minute, to defend the address, plus one bubble exchange at the beginning of each session with a new peer; adding all headers, the bubble size is about 144 bytes, which results in about 20 bps of traffic on the server. In short, the amount of traffic seen by the Teredo server is 250 times less that the traffic seen by a Teredo client.

The tunnel approach is more expensive to provide, because the tunnel server will have to carry a much larger amount of traffic. It is unclear that a tunnel service can be provided as an almost free "supporting infrastructure", except perhaps if the service was directly provided by the same ISP that already provides IPv4 connectivity to the unmanaged network.

The two approaches have much in common: both need to parameterize a client with the name of a server and with sufficient credentials; the encapsulation is similar; both use the same encapsulation mechanism; and both will use router solicitations to obtain an address prefix. The main difference is the handling of bubbles in Teredo, which are used to defend the address and to initiate sessions with peers. This is obviously more complex than sending the packet without any processing, but the complexity is only marginally

higher than the discovery of link layer addresses using neighbor discovery. In short, Teredo is more complex, but the complexity is not overwhelming.

An advantage of the tunnel server approach over Teredo is that it will work regardless of the type and number of NAT between the client and the tunnel server. In contrast, Teredo will not work across a "symmetric" NAT, and communication may be impossible between two Teredo clients located behind the same NAT.

Another potential advantage of the tunnel server approach is that it could provide clients with stable IPv6 addresses. This is only a potential advantage, since the server may prefer to delegate addresses on a session per session basis, or may want to operate in a stateless manner.

2.1.4 Recommendation

Teredo appears to be a good fit for providing IPv6 connectivity to hosts behind NAT, in case A of IPv6 deployment. The service is designed for minimizing the cost of deploying the server, which matches the requirement of minimizing the cost of the "supporting infrastructure" for peer-to-peer applications.

There are two situations in which a tunnel service makes more sense: when the cost of bandwidth is not a concern, and when the unmanaged network is located behind a symmetric NAT. In these two cases, using a tunnel service is preferred over using Teredo.

The most reasonable solution would be to develop a tunnel service specification that is compatible with Teredo, so that a given host could be configured to use either Teredo or the tunnel service, depending on the server configured in the dual stack host.

2.2 Security considerations in case A

A characteristic of case A is that a host located behind a NAT acquires global IPv6 connectivity, using either Teredo or an alternative tunneling mechanism. If no precaution is taken, there is a risk of exposing to the global Internet some applications and services that are only expected to serve local hosts, located behind the NAT. Developers and administrators should make sure that the global IPv6 connectivity is restricted to only those applications that are expressly designed for global Internet connectivity.

3 Meeting case B requirements

In case B, we assume that the gateway and the ISP are both dual stack. The hosts on the local network may be IPv4 only, dual stack, or IPv6 only. The main requirements are: prefix delegation, and name resolution. We also study the potential need for communication between IPv4 and IPv6 hosts, and conclude that a dual stack approach

is preferable.

3.1 Prefix delegation

The gateway must be able to acquire an IPv6 prefix, delegated by the ISP. The possible mechanisms are RA proxy and explicit prefix delegation.

3.1.1 RA proxy

The implicit delegation mechanism assumes that the gateway is connected to the ISP by a point-to-point link. Examples of such point to point links are various types of configured tunnels and serial links, for example using PPP. The principle of RA proxy is simple: the gateway issues a "router solicitation" message on the serial link, receives a "router advertisement", learns a network prefix from the advertisement, and advertises the same prefix on the unmanaged network.

The RA proxy method results in the sharing of the same prefix over several links, a procedure generally known as "multi-link subnet". This sharing has effects on neighbor discovery protocols, and possibly also on other protocols such as LLMNR that rely on "link local multicast". These effects need to be carefully studied.

3.1.2 Explicit prefix delegation

Several networks have already started using an explicit prefix delegation mechanism using DHCPv6 [PREFIXDHCPV6]. In this mechanism, the gateway uses a DHCP request to obtain an adequate prefix from a

DHCP server managed by the Internet Service Provider. The DHCP request is expected to carry proper identification of the gateway, which enables the ISP to implement prefix delegation policies.

The basic use of DHCP is insecure. This may be a problem if the link between gateway and ISP is shared by multiple subscribers. In that case, some security procedure will have to be used, to ensure at a minimum that DHCP requests and replies are properly authenticated.

3.1.3 Recommendation

The RA proxy and DHCP methods appear to have different domains of application. RA proxy is a simple method that corresponds well to "informal sharing" of a link, while explicit delegation provides strong administrative control. Both methods require development: specify the interaction with neighbor discovery for RA proxy; provide security guidelines for explicit delegation. Proceeding with standardization of at least one method, and preferably both, is quite urgent.

3.2 Communication between IPv4-only and IPv6-only hosts

During the transition phase from IPv4 to IPv6 there will be IPv4-only, dual stack and IPv6-only nodes. In theory, there may be a need to provide some interconnection services so that IPv4-only and IPv6-only hosts can communicate. However, as indicated in a companion document [NATPTBISSUES], it is hard to develop a translation service that does not have unwanted side effects on the efficiency or the security of communications. As a consequence, the authors recommend that, if a device has a requirement to communicate with IPv4 only hosts, this device implements an IPv4 stack. The only devices that should only have IPv6 connectivity are those that are intended to only communicate with IPv6 hosts.

3.3 Resolution of names to IPv6 addresses

There are three types of name resolution services that should be provided in case B: local IPv6 capable hosts must be able to obtain the IPv6 addresses of correspondent hosts on the Internet; they should be able to publish their address if they want to be accessed from the Internet; and they should be able to obtain the IPv6 address of other local IPv6 hosts. These three problems are described in the next sections.

3.3.1 Provisioning the address of a DNS resolver

In an unmanaged environment, IPv4 hosts usually obtain the address of the local DNS resolver through DHCPv4; the DHCPv4 service is generally provided by the gateway. The gateway will also use DHCPv4 to obtain the address of a suitable resolver from the local Internet service provider.

The DHCPv4 solution will suffice in practice for the gateway and also for the dual stack hosts. There is evidence that even the simple DNS resolvers present in small gateways can relay arbitrary DNS request and serve arbitrary DNS records, including AAAA records.

Just using DHCPv4 will not be an adequate solution for IPv6 only local hosts. Three solutions have been envisaged for these hosts: either using DHCPv6 to obtain the address of the DNS server [DNSDHCPV6]; sending the DNS requests to a well known IPv6 address [DNSANYCAST]; or sending the DNS requests to the IPv6 address of the gateway itself.

3.3.2 Publishing IPv6 addresses to the Internet

IPv6 capable hosts may be willing to provide services accessible from the global Internet. They will thus need to document their address in a server that is publicly available. IPv4 hosts in unmanaged networks have a similar problem today, which they solve using one of three possible solutions:

* Manual configuration of a stable address in a DNS server;
* Dynamic configuration using the standard dynamic DNS protocol;

* Dynamic configuration using an ad hoc protocol.

Manual configuration of stable addresses is not satisfactory in an unmanaged IPv6 network: the prefix allocated to the gateway may or may not be stable, and in any case copying long binary addresses through a manual procedure is error prone.

Dynamic configuration using the same type of ad hoc protocols that are common today is indeed possible, but the IETF should encourage the use of standard solutions based on DDNS.

3.3.3 Resolving local IPv6 addresses

There are two possible ways of resolving local IPv6 addresses: one may either publish the IPv6 addresses in a local server, or one may use a peer-to-peer address resolution protocol such as link local multicast name resolution [LLMNR].

The use of a local server requires that IPv6 capable hosts discover this server, as explained in 3.3.1, and then that they use a protocol such as DDNS to publish their IPv6 addresses to this server. In practice, the DNS address discovered in 3.3.1 will often be the address of the gateway itself, and the local server will thus be the gateway. Implementing a dynamic DNS server on the gateway may be problematic, as many of these gateways are very small devices with limited memory and limited processing power.

An alternative to using a local server is LLMNR, which uses a multicast mechanism to resolve DNS requests. LLMNR does not require any service from the gateway, and also does not require that hosts use DDNS. LLMNR relies on multicast for its operation. There are scaling issues with using multicast, as the procedure may become very chatty in large networks; but this is not a practical problem in most unmanaged networks. A more important problem is that some networks only have limited support for multicast transmission: for example, multicast transmission on 802.11 network is error prone. However, unmanaged networks also use multicast for neighbor discovery; the requirements of ND and LLMNR are similar; if a link technology supports use of ND, it will also enable use of LLMNR.

3.3.4 Recommendations for name resolution

The IETF should quickly provide a recommended procedure for provisioning the DNS resolver in IPv6 only hosts, either by standardizing the proper DHCPv6 subset, or by recommending an alternate convention.

The most plausible candidate for local name resolution appears to be LLMNR; the IETF should quickly proceed to the standardization of that protocol.

3.4 Security considerations for case B

The case B solutions provide global IPv6 connectivity to the local hosts. Removing the limit to connectivity imposed by NAT is both a feature and a risk. Implementations should carefully limit global IPv6 connectivity to only those applications that are specifically designed to operate on the global Internet.

4 Meeting case C requirements

Case C is very similar to case B, the difference being that the ISP is not dual stack. The gateway must thus use some form of tunneling mechanism to obtain IPv6 connectivity, and an address prefix.

4.1 Tunneling mechanisms

4.1.1 6to4

The [6TO4] technology allows routers to derive a global scope IPv6 prefix from a global IPv4 address. This technology is a very good fit for the second phase of the transition, as it can be programmed in the "upgraded gateway", and can provide value to the gateway users without requiring explicit support from the ISP. This technology has however a clear limitation: it requires that the gateway obtains at least one global IPv4 address from the local ISP.

Another potential limitation of the technology is the reliance on publicly accessible "6to4 relay routers" that accept packets from 6to4 routers and relay them to the "regular" IPv6 Internet. These relays all listen to the same IPv4 anycast address [RFC3056], which enables gateways to start operating as 6to4 routers without requiring any explicit configuration. As the deployment of IPv6 progresses, a growing fraction of the traffic originating from 6to4 routers will have to be carried through these relays, potentially leading to severe congestion of the relays.

There are three possible ways to alleviate this congestion. First, one can hope that many actors will deploy 6to4 relay routers, in order to facilitate the deployment of IPv6; congestion would be alleviated by the provision of a large number of gateways. Second, one could develop some "route optimization" process, so that the traffic would flow through a "shortcut path" rather than through the 6to4 relays; the relays would then avoid congestion by carrying only a small fraction of the traffic. Third, if neither the first nor the second solution materializes, some gateways may enter into contractual agreements with relay service providers; in this case, the 6to4 technology would become merely a variant of the configured tunnel technologies.

4.1.2 Tunnel broker

Configured tunnels require a contractual agreement with an IPv6 provider, which comes in addition to the existing agreement with the

IPv4 provider; different technologies have different domains of application:

- Many tunnel technologies use a global IPv4 address to identify the "client end" of the tunnel, thus inheriting the same "global IPv4 address" requirement as 6TO4;

- A variant of the [TEREDO] technology could be used to establish tunnels over UDP when the client is behind a NAT; this variant is however not standardized.

- Practical deployment of tunnel technologies requires the introduction of accounting/billing functions; the existing tunnel broker specification, [TUNNELS], does not describe how these functions should be implemented. (The use of public relays in the 6to4 technology may raise a similar issue.)

Configured tunnels are in practice an intermediate solution between the "automatic configuration" provided by 6to4, and the "ISP support" that characterizes case B.

4.1.3 Recommendations

The practical conclusion of the previous analysis is that "upgraded gateways" will probably support the 6TO4 technology, and will have an optional configuration option for "configured tunnels".

The tunnel broker technology should be augmented, to include support for accounting and billing functions.

Due to concerns with potential overload of public 6to4 relays, the 6to4 implementations should include a configuration option that let user take advantage of specific relays.

4.2 Security considerations

Providing local hosts with global IPv6 connectivity raises the same issues in case C as in case B. Discussions in the V6OPS working group have also raised a potential security concern with the 6to4 solution: an attacker can send a packet with a spoofed IPv6 address to a host on the unmanaged network; the host will reply to the source address; the reply traffic will be hard to trace back to the original attacker.

This attack is only an additional vulnerability if IPv4 source addresses cannot be spoofed; there is thus a debate between those who believe that we will eventually make IPv4 addresses hard to spoof, and those who believe that this goal will never be achieved. Actually thwarting the attack would require a slight addition to the 6to4 specification: 6to4 routers that forward a encapsulated packet to a native IPv6 network should implement a form of "traceback" mechanism.

5 Meeting the case D requirements

In case D, the ISP only provides IPv6 services.

5.1.1 IPv6 addressing requirements

We expect IPv6 addressing in case D to proceed similarly to case B, i.e. use either RA proxy or explicit configuration to provision an IPv6 prefix on the gateway.

5.1.2 **IPv4** connectivity requirements

Local IPv4 capable hosts may want to still access IPv4 only services. The proper way to do this for dual stack nodes in the unmanaged network is to develop a form of "IPv4 over IPv6" tunneling. This tunneling protocol need to be standardized, possibly as an extension of the "dual stack transition mechanism" proposal [DSTM]. Part of the standardization will have to cover configuration issues, i.e. how to provision the IPv4 capable hosts with the address of the local IPv4 tunnel servers.

5.1.3 Naming requirements

Naming requirements are similar to case B, with one difference: the gateway cannot expect to use DHCPv4 to obtain the address of the DNS resolver recommended by the ISP.

6 Provisional recommendations

This draft is still a draft, but we can already list a set of recommendations for the V6OPS working group:

- To meet case A requirements, we need to develop and standardize the Teredo or similar technology.

- To meet case B prefix delegation requirements, we need a standardized IPv6 prefix delegation mechanism

- To meet case B naming requirements, we need to standardize a way to provision a DNS resolver address in IPv6 only hosts, and we need to proceed with the standardization fo LLMNR.

- To meet case C connectivity requirement, we need to continue standardization of the 6to4 mechanism.

- To meet case D IPv4 connectivity requirement, we need to standardize an IPv4 over IPv6 tunneling mechanism, as well as the associated configuration services.

7 IANA Considerations

This memo does not include any request to IANA.

8 Copyright

The following copyright notice is copied from RFC 2026 [Bradner, 1996], Section 10.4, and describes the applicable copyright for this document.

9 Intellectual Property

The following notice is copied from RFC 2026 [Bradner, 1996], Section 10.4, and describes the position of the IETF concerning intellectual property claims made against this document.

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use other technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such

10 Acknowledgements

This fractional and preliminary memo has benefited from comments of Margaret Wasserman and Tony Hain.

11 References

[UNMANREQ] Unmanaged Networks Transition Scope. Work in progress.

[NATPTBISSUES] Issues when translating between IPv4 and IPv6. Work in progress.

[IPV6] S. Deering, R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[NEIGHBOR] T. Narten, E. Nordmark, W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 2461, December 1998.

[STATELESS] T. Narten, S. Thomson, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.

[6TO4] B. Carpenter, K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, February 2001.

[6TO4ANYCAST] C. Huitema, "An Anycast Prefix for 6to4 Relay Routers", RFC 3068, June 2001.

[TEREDO] Teredo: Tunneling IPv6 over UDP through NATs. Work in progress.

[TUNNELS] A. Durand, P. Fasano, I. Guardini. IPv6 Tunnel Broker. RFC 3053, January 2001

[PREFIXDHCPV6] IPv6 Prefix Options for DHCPv6. Work in progress.

[DNSDHCPV6] DNS Configuration options for DHCPv6. Work in progress.

[DNSANYCAST] Well known site local unicast addresses for DNS resolver. Work in progress.

[LLMNR] Link Local Multicast Name Resolution. Work in progress.

[DSTM] Dual Stack Transition Mechanism (DSTM). Work in progress.

12 Authors' Addresses

Christian Huitema
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399
Email: huitema@microsoft.com

Rob Austein
Email: sra@hactrn.net

Suresh Satapati
Cisco Systems, Inc.
San Jose, CA 95134
USA
EMail: satapati@cisco.com

Ronald van der Pol
Email: Ronald.vanderPol@rvdp.org

Table of Contents: