Network Working Group
Request for Comments: 3904
Category: Informational

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

September 2004

Evaluation of IPv6 Transition Mechanisms for Unmanaged Networks

Status of this Memo

Copyright Notice

Abstract

This document analyzes issues involved in the transition of "unmanaged networks" from IPv4 to IPv6. Unmanaged networks typically correspond to home networks or small office networks. A companion paper analyzes out the requirements for mechanisms needed in various transition scenarios of these networks to IPv6. Starting from this analysis, we evaluate the suitability of mechanisms that have already been specified, proposed, or deployed.

Table of Contents:

1. Introduction

This document analyzes the issues involved in the transition from IPv4 to IPv6 [IPV6]. In a companion paper [UNMANREQ] we defined the "unmanaged networks", which typically correspond to home networks or small office networks, and the requirements for transition mechanisms in various scenarios of transition to IPv6.

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

A) a gateway which does not provide IPv6 at all;
B) a dual-stack gateway connected to a dual-stack ISP;
C) a dual-stack gateway connected to an IPv4-only ISP; and
D) a gateway connected to an IPv6-only ISP.

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.

The issues involved are described in the next sections. This analysis outlines two types of requirements: 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. The connectivity requirements often require tunneling solutions. We devote the first section of this memo to an evaluation of various tunneling solutions.

2. Evaluation of Tunneling Solutions

In the case A and case C scenarios described in [UNMANREQ], the unmanaged network cannot obtain IPv6 service, at least natively, from its ISP. In these cases, the IPv6 service will have to be provided through some form of tunnel. There have been multiple proposals on different ways to tunnel IPv6 through an IPv4 service. We believe that these proposals can be categorized according to two important properties:

* Is the deployment automatic, or does it require explicit configuration or service provisioning?

* Does the proposal allow for the traversal of a NAT?

These two questions divide the solution space into four broad classes. Each of these classes has specific advantages and risks, which we will now develop.

2.1. Comparing Automatic and Configured Solutions

It is possible to broadly classify tunneling solutions as either "automatic" or "configured". In an automatic solution, a host or a router builds an IPv6 address or an IPv6 prefix by combining a pre-defined prefix with some local attribute, such as a local IPv4 address [6TO4] or the combination of an address and a port number [TEREDO]. Another typical and very important characteristic of an automatic solution is they aim to work with a minimal amount of support or infrastructure for IPv6 in the local or remote ISPs.

In a configured solution, a host or a router identifies itself to a tunneling service to set up a "configured tunnel" with an explicitly defined "tunnel router". The amount of actual configuration may vary from manually configured static tunnels to dynamic tunnel services requiring only the configuration of a "tunnel broker", or even a completely automatic discovery of the tunnel router.

Configured tunnels have many advantages over automatic tunnels. The client is explicitly identified and can obtain a stable IPv6 address. The service provider is also well identified and can be held responsible for the quality of the service. It is possible to route multicast packets over the established tunnel. There is a clear address delegation path, which enables easy support for reverse DNS lookups.

Automatic tunnels generally cannot provide the same level of service. The IPv6 address is only as stable as the underlying IPv4 address, the quality of service depends on relays operated by third parties, there is typically no support for multicast, and there is often no easy way to support reverse DNS lookups (although some workarounds are probably possible). However, automatic tunnels have other advantages. They are obviously easier to configure, since there is no need for an explicit relation with a tunnel service. They may also be more efficient in some cases, as they allow for "path optimization".

2.1.1. Path Optimization in Automatic Tunnels

In automatic tunnels like [TEREDO] and [6TO4], the bulk of the traffic between two nodes using the same technology is exchanged on a direct path between the endpoints, using the IPv4 services to which the endpoints already subscribe. By contrast, the configured tunnel servers carry all the traffic exchanged by the tunnel client.

Path optimization is not a big issue if the tunnel server is close to the client on the natural path between the client and its peers. However, if the tunnel server is operated by a third party, this third party will have to bear the cost of provisioning the bandwidth used by the client. The associated costs can be significant.

These costs are largely absent when the tunnels are configured by the same ISP that provides the IPv4 service. The ISP can place the tunnel end-points close to the client, i.e., mostly on the direct path between the client and its peers.

2.1.2. Automatic Tunnels and Relays

The economics arguments related to path optimization favor either configured tunnels provided by the local ISP or automatic tunneling regardless of the co-operation of ISPs. However, automatic solutions require that relays be configured throughout the Internet. If a host that obtained connectivity through an automatic tunnel service wants to communicate with a "native" host or with a host using a configured

tunnel, it will need to use a relay service, and someone will have to provide and pay for that service. We cannot escape economic considerations for the deployment of these relays.

It is desirable to locate these relays close to the "native host". During the transition period, the native ISPs have an interest in providing a relay service for use by their native subscribers. Their subscribers will enjoy better connectivity, and will therefore be happier. Providing the service does not result in much extra bandwidth requirement: the packets are exchanged between the local subscribers and the Internet; they are simply using a v6-v4 path instead of a v6-v6 path. (The native ISPs do not have an incentive to provide relays for general use; they are expected to restrict access to these relays to their customers.)

We should note however that different automatic tunneling techniques have different deployment conditions.

2.1.3. The Risk of Several Parallel IPv6 Internets

In an early deployment of the Teredo service by Microsoft, the relays are provided by the native (or 6to4) hosts themselves. The native or 6to4 hosts are de-facto "multi-homed" to native and Teredo hosts, although they never publish a Teredo address in the DNS or otherwise. When a native host communicates with a Teredo host, the first packets are exchanged through the native interface and relayed by the Teredo server, while the subsequent packets are tunneled "end-to-end" over IPv4 and UDP. This enables deployment of Teredo without having to field an infrastructure of relays in the network.

This type of solution carries the implicit risk of developing two parallel IPv6 Internets, one native and one using Teredo: in order to communicate with a Teredo-only host, a native IPv6 host has to implement a Teredo interface. The Teredo implementations try to mitigate this risk by always preferring native paths when available, but a true mitigation requires that native hosts do not have to implement the transition technology. This requires cooperation from the IPv6 ISP, who will have to support the relays. An IPv6 ISP that really wants to isolate its customers from the Teredo technology can do that by providing native connectivity and a Teredo relay. The ISP’s customers will not need to implement their own relay.

Communication between 6to4 networks and native networks uses a different structure. There are two relays, one for each direction of communication. The native host sends its packets through the nearest 6to4 router, i.e., the closest router advertising the 2002::/16 prefix through the IPv6 routing tables; the 6to4 network sends its packet through a 6to4 relay that is either explicitly configured or

discovered through the 6to4 anycast address 192.88.99.1 [6TO4ANYCAST]. The experience so far is that simple 6to4 routers are easy to deploy, but 6to4 relays are scarce. If there are too few relays, these relays will create a bottleneck. The communications between 6to4 and native networks will be slower than the direct communications between 6to4 hosts. This will create an incentive for native hosts to somehow "multi-home" to 6to4, de facto creating two parallel Internets, 6to4 and native. This risk will only be mitigated if there is a sufficient deployment of 6to4 relays.

The configured tunnel solutions do not carry this type of risk.

2.1.4. Lifespan of Transition Technologies

A related issue is the lifespan of the transition solutions. Since automatic tunneling technologies enable an automatic deployment, there is a risk that some hosts never migrate out of the transition. The risk is arguably less for explicit tunnels: the ISPs who provide the tunnels have an incentive to replace them with a native solution as soon as possible.

Many implementations of automatic transition technologies incorporate an "implicit sunset" mechanism: the hosts will not configure a transition technology address if they have native connectivity; the address selection mechanisms will prefer native addresses when available. The transition technologies will stop being used eventually, when native connectivity has been deployed everywhere. However, the "implicit sunset" mechanism does not provide any hard guarantee that transition will be complete at a certain date.

Yet, the support of transition technologies has a cost for the entire network: native IPv6 ISPS have to support relays in order to provide good performance and avoid the "parallel Internet" syndrome. These costs may be acceptable during an initial deployment phase, but they can certainly not be supported for an indefinite period. The "implicit sunset" mechanisms may not be sufficient to guarantee a finite lifespan of the transition.

2.2. Cost and Benefits of NAT Traversal

During the transition, some hosts will be located behind IPv4 NATs. In order to participate in the transition, these hosts will have to use a tunneling mechanism designed to traverse NAT.

We may ask whether NAT traversal should be a generic property of any transition technology, or whether it makes sense to develop two types of technologies, some "NAT capable" and some not. An important question is also which kinds of NAT boxes one should be able to

traverse. One should probably also consider whether it is necessary to build an IPv6 specific NAT traversal mechanism, or whether it is possible to combine an existing IPv4 NAT traversal mechanism with some form of IPv6 in IPv4 tunneling. There are many IPv4 NAT traversal mechanisms; thus one may ask whether these need re-invention, especially when they are already complex.

A related question is whether the NAT traversal technology should use automatic tunnels or configured tunnels. We saw in the previous section that one can argue both sides of this issue. In fact, there are already deployed automatic and configured solutions, so the reality is that we will probably see both.

2.2.1. Cost of NAT Traversal

NAT traversal technologies generally involve encapsulating IPv6 packets inside a transport protocol that is known to traverse NAT, such as UDP or TCP. These transport technologies require significantly more overhead than the simple tunneling over IPv4 used in 6to4 or in IPv6 in IPv4 tunnels. For example, solutions based on UDP require the frequent transmission of "keep alive" packets to maintain a "mapping" in the NAT; solutions based on TCP may not require such a mechanism, but they incur the risk of "head of queue blocking", which may translate in poor performance. Given the difference in performance, it makes sense to consider two types of transition technologies, some capable of traversing NAT and some aiming at the best performance.

2.2.2. Types of NAT

There are many kinds of NAT on the market. Different models implement different strategies for address and port allocations, and different types of timers. It is desirable to find solutions that cover "almost all" models of NAT.

A configured tunnel solution will generally make fewer hypotheses on the behavior of the NAT than an automatic solution. The configured solutions only need to establish a connection between an internal node and a server; this communication pattern is supported by pretty much all NAT configurations. The variability will come from the type of transport protocols that the NAT supports, especially when the NAT also implements "firewall" functions. Some models will allow establishment of a single "protocol 41" tunnel, while some may prevent this type of transmission. Some models will allow UDP transmission, while other may only allow TCP, or possibly HTTP.

The automatic solutions have to rely on a "lowest common denominator" that is likely to be accepted by most models of NAT. In practice, this common denominator is UDP. UDP based NAT traversal is required by many applications, e.g., networked games or voice over IP. The experience shows that most recent "home routers" are designed to support these applications. In some edge cases, the automatic solutions will require explicit configuration of a port in the home router, using the so-called "DMZ" functions; however, these functions are hard to use in an "unmanaged network" scenario.

2.2.3. Reuse of Existing Mechanisms

NAT traversal is not a problem for IPv6 alone. Many IPv4 applications have developed solutions, or kludges, to enable communication across a NAT.

Virtual Private Networks are established by installing tunnels between VPN clients and VPN servers. These tunnels are designed today to carry IPv4, but in many cases could easily carry IPv6. For example, the proposed IETF standard, L2TP, includes a PPP layer that can encapsulate IPv6 as well as IPv4. Several NAT models are explicitly designed to pass VPN traffic, and several VPN solutions have special provisions to traverse NAT. When we study the establishment of configured tunnels through NAT, it makes a lot of sense to consider existing VPN solutions.

[STUN] is a protocol designed to facilitate the establishment of UDP associations through NAT, by letting nodes behind NAT discover their "external" address. The same function is required for automatic tunneling through NAT, and one could consider reusing the STUN specification as part of an automatic tunneling solution. However, the automatic solutions also require a mechanism of bubbles to establish the initial path through a NAT. This mechanism is not present in STUN. It is not clear that a combination of STUN and a bubble mechanism would have a technical advantage over a solution specifically designed for automatic tunneling through NAT.

2.3. Development of Transition Mechanisms

The previous sections make the case for the development of four transition mechanism, covering the following 4 configurations:

- Configured tunnel over IPv4 in the absence of NAT;
- Automatic tunnel over IPv4 in the absence of NAT;
- Configured tunnel across a NAT;
- Automatic tunnel across a NAT.

Teredo is an example of an already designed solution for automatic tunnels across a NAT; 6to4 is an example of a solution for automatic tunnels over IPv4 in the absence of NAT.

All solutions should be designed to meet generic requirements such as security, scalability, support for reverse name lookup, or simple management. In particular, automatic tunneling solutions may need to be augmented with a special purpose reverse DNS lookup mechanism, while configured tunnel solutions would benefit from an automatic service configuration mechanism.

3. Meeting Case A Requirements

In case A, isolated hosts need to acquire some form of 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.

3.1. Evaluation of Connectivity Mechanisms

In case A, IPv6 capable hosts seek IPv6 connectivity in order to communicate with applications in the global IPv6 Internet. The connectivity requirement can be met using either configured tunnels or automatic tunnels.

If the host is located behind a NAT, the tunneling technology should be designed to traverse NAT; tunneling technologies that do not support NAT traversal can obviously be used if the host is not located behind a NAT.

When the local ISP is willing to provide a configured tunnel solution, we should make it easy for the host in case A to use it. The requirements for such a service will be presented in another document.

An automatic solution like 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".

3.2. Security Considerations in Case A

A characteristic of case A is that an isolated host 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, e.g., those located behind the NAT

when a NAT is present. 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. The users should be able to configure which applications get IPv6 connectivity to the Internet and which should not.

Any solution to the NAT traversal problem is likely to involve relays. There are concerns that improperly designed protocols or improperly managed relays could open new avenues for attacks against Internet services. This issue should be addressed and mitigated in the design of the NAT traversal protocols and in the deployment guides for relays.

4. 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.

4.1. Connectivity

The gateway must be able to acquire an IPv6 prefix, delegated by the ISP. This can be done through explicit prefix delegation (e.g., [DHCPV6, PREFIXDHCPV6]), or if the ISP is advertising a /64 prefix on the link, such a link can be extended by the use of an ND proxy or a bridge.

An ND proxy can also be used to extend a /64 prefix to multiple physical links of different properties (e.g., an Ethernet and a PPP link).

4.1.1. Extending a Subnet to Span Multiple Links

A /64 subnet can be extended to span multiple physical links using a bridge or ND proxy. Bridges can be used when bridging multiple similar media (mainly, Ethernet segments). On the other hand, an ND proxy must be used if a /64 prefix has to be shared across media (e.g., an upstream PPP link and a downstream Ethernet), or if an interface cannot be put into promiscuous mode (e.g., an upstream wireless link).

Extending a single subnet to span from the ISP to all of the unmanaged network is not recommended, and prefix delegation should be used when available. However, sometimes it is unavoidable. In

addition, sometimes it's necessary to extend a subnet in the unmanaged network, at the "customer-side" of the gateway, and changing the topology using routing might require too much expertise.

The ND 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 [LLMNR] that rely on "link local multicast". These effects need to be carefully studied.

4.1.2. Explicit Prefix Delegation

Several networks have already started using an explicit prefix delegation mechanism using DHCPv6. 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. It is expected that the ISP assigns a /48 to the customer. The gateway should automatically assign /64s out of this /48 to its internal links.

DHCP is insecure unless authentication is used. This may be a particular problem if the link between gateway and ISP is shared by multiple subscribers. DHCP specification includes authentication options, but the operational procedures for managing the keys and methods for sharing the required information between the customer and the ISP are unclear. To be secure in such an environment in practice, the practical details of managing the DHCP authentication need to be analyzed.

4.1.3. Recommendation

The ND proxy and DHCP methods appear to have complementary domains of application. ND proxy is a simple method that corresponds well to the "informal sharing" of a link, while explicit delegation provides strong administrative control. Both methods require development: specify the interaction with neighbor discovery for ND proxy; provide security guidelines for explicit delegation.

4.2. Communication Between IPv4-only and IPv6-capable Nodes

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, 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 requires communication with

IPv4-only hosts, this device implements an IPv4 stack. The only devices that should have IPv6-only connectivity are those that are intended to only communicate with IPv6 hosts.

4.3. 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. Operational considerations and issues with IPv6 DNS are analyzed in [DNSOPV6].

4.3.1. 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 DNS servers accessed over IPv4 can serve arbitrary DNS records, including AAAA records.

Just using DHCPv4 will not be an adequate solution for IPv6-only local hosts. The DHCP working group has defined how to use (stateless) DHCPv6 to obtain the address of the DNS server [DNSDHCPV6]. DHCPv6 and several other possibilities are being looked at in the DNSOP Working Group.

4.3.2. 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 publish 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 hexadecimal strings 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 Dynamic DNS (DDNS).

4.3.3. Resolving the IPv6 Addresses of Local Hosts

There are two possible ways of resolving the IPv6 addresses of local hosts: one may either publish the IPv6 addresses in a DNS server for the local domain, or one may use a peer-to-peer address resolution protocol such as LLMNR.

When a DNS server is used, this server could in theory be located anywhere on the Internet. There is however a very strong argument for using a local server, which will remain reachable even if the network connectivity is down.

The use of a local server requires that IPv6 capable hosts discover this server, as explained in 4.3.1, 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 4.3.1 will often be the address of the gateway itself, and the local server will thus be the gateway.

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. An 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 [NEIGHBOR]; the requirements of ND and LLMNR are similar; if a link technology supports use of ND, it can also enable use of LLMNR.

4.3.4. Recommendations for Name Resolution

The IETF should quickly provide a recommended procedure for provisioning the DNS resolver in IPv6-only hosts.

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

4.4. Security Considerations in 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. Local applications, for example, could be restricted to only use link-local addresses, or addresses whose most significant bits match the prefix of the local subnet, e.g., a prefix advertised as "on link" in a local router advertisement. There is a debate as to whether such restrictions should be "per-site" or "per-link", but this is not a serious issue when an unmanaged network is composed of a single link.

5. 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.

A simplified form of case B is a single host with a global IPv4 address, i.e., with a direct connection to the IPv4 Internet. This host will be able to use the same tunneling mechanisms as a gateway.

5.1. Connectivity

Connectivity in case C requires some form of tunneling of IPv6 over IPv4. The various tunneling solutions are discussed in section 2.

The requirements of case C can be solved by an automatic tunneling mechanism such as 6to4 [6TO4]. An alternative may be the use of a configured tunnels mechanism [TUNNELS], but as the local ISP is not IPv6-enabled, this may not be feasible. The practical conclusion of our 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 some form of automatic configuration.

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

6. Meeting the Case D Requirements

In case D, the ISP only provides IPv6 services.

6.1. IPv6 Addressing Requirements

We expect IPv6 addressing in case D to proceed similarly to case B, i.e., use either an ND proxy or explicit prefix delegation through DHCPv6 to provision an IPv6 prefix on the gateway.

6.2. IPv4 Connectivity Requirements

Local IPv4 capable hosts may still want to 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. There are no standardized solutions and the IETF has devoted very little effort to this issue, although there is ongoing work with [DSTM] and [TSP]. A solution needs to be standardized. 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.

6.3. 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.

7. Recommendations

After a careful analysis of the possible solutions, we can list a set of recommendations for the V6OPS working group:

1. To meet case A and case C requirements, we need to develop, or continue to develop, four types of tunneling technologies: automatic tunnels without NAT traversal such as [6TO4], automatic tunnels with NAT traversal such as [TEREDO], configured tunnels without NAT traversal such as [TUNNELS, TSP], and configured tunnels with NAT traversal.

2. To facilitate the use of configured tunnels, we need a standardized way for hosts or gateways to discover the tunnel server or tunnel broker that may have been configured by the local ISP.

3. To meet case B "informal prefix sharing" requirements, we would need a standardized way to perform "ND proxy", possibly as part of a "multi-link subnet" specification. (The explicit prefix delegation can be accomplished through [PREFIXDHCPV6].)

4. To meet case B naming requirements, we need to proceed with the standardization of LLMNR. (The provisioning of DNS parameters can be accomplished through [DNSDHCPV6].)

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

8. Security Considerations

This memo describes the general requirements for transition mechanisms. Specific security issues should be studied and addressed during the development of the specific mechanisms.

When hosts which have been behind a NAT are exposed to IPv6, the security assumptions may change radically. This is mentioned in sections 3.2 and 4.4. One way to cope with that is to have a default firewall with a NAT-like access configuration; however, any such firewall configuration should allow for easy authorization of those applications that actually need global connectivity. One might also restrict applications which can benefit from global IPv6 connectivity on the nodes.

Security policies should be consistent between IPv4 and IPv6. A policy which prevents use of v6 while allowing v4 will discourage migration to v6 without significantly improving security. Developers and administrators should make sure that global Internet connectivity through either IPv4 or IPv6 is restricted to only those applications that are expressly designed for global Internet connectivity.

Several transition technologies require relays. There are concerns that improperly designed protocols or improperly managed relays could open new avenues for attacks against Internet services. This issue should be addressed and mitigated in the design of the transition technologies and in the deployment guides for relays.

9. Acknowledgements

This memo has benefited from the comments of Margaret Wasserman, Pekka Savola, Chirayu Patel, Tony Hain, Marc Blanchet, Ralph Droms, Bill Sommerfeld, and Fred Templin. Tim Chown provided a lot of the analysis for the tunneling requirements work.

10. References

10.1. Normative References

[UNMANREQ] Huitema, C., Austein, R., Satapati, S., and R. van der Pol, "Unmanaged Networks IPv6 Transition Scenarios", RFC 3750, April 2004.

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

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

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

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

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

[DHCPV6] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[DNSDHCPV6] Droms, R., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, December 2003.

[PREFIXDHCPV6] Troan, O. and R. Droms, "IPv6 Prefix Options for Dynamic Host Configuration Protocol (DHCP) version 6", RFC 3633, December 2003.

10.2. Informative References

[STUN] Rosenberg, J., Weinberger, J., Huitema, C., and R. Mahy, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)", RFC 3489, March 2003.

[DNSOPV6] Durand, A., Ihren, J., and P. Savola. "Operational Considerations and Issues with IPv6 DNS", Work in Progress.

[LLMNR] Esibov, L., Aboba, B., and D. Thaler, "Linklocal Multicast Name Resolution (LLMNR)", Work in Progress.

[TSP] Blanchet, M., "IPv6 Tunnel Broker with the Tunnel Setup Protocol(TSP)", Work in Progress.

[DSTM] Bound, J., "Dual Stack Transition Mechanism", Work in Progress.

[TEREDO] Huitema, C., "Teredo: Tunneling IPv6 over UDP through NATs", Work in Progress.

11. Authors' Addresses

Christian Huitema
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

EMail: huitema@microsoft.com

Rob Austein
Internet Systems Consortium
950 Charter Street
Redwood City, CA 94063
USA

EMail: sra@isc.org

Suresh Satapati
Cisco Systems, Inc.
San Jose, CA 95134
USA

EMail: satapati@cisco.com

Ronald van der Pol
NLnet Labs
Kruislaan 419
1098 VA Amsterdam
NL

EMail: Ronald.vanderPol@nlnetlabs.nl

12. Full Copyright Statement

Intellectual Property

Acknowledgement

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