behave Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 15, 2010

C. Perkins, Ed.
WiChorus Inc.

October 12, 2009

Translating IPv4 to IPv6 based on source IPv4 address
draft-perkins-sourceipnat-01

Status of this Memo

Copyright Notice

Abstract

A method is proposed to enable communications between an IPv4-only node in today's Internet and an IPv6-only node, initiated by the IPv4-only node. The communication depends on allocation of a flow record and address triggered by a DNS query received for the target v6-only node. DNS query conventions can be agreed upon to provide a natural model for resolving IPv4 queries for IPv6-only nodes. The NAT mechanism proposed demultiplexes multiple sessions through the same dynamically allocated IP address, using flow records matching the source address of incoming packets. This is in contrast to the use of ports in NAT-PT boxes, which inhibits the support of incoming traffic towards a node behind the NAT-PT.

Table of Contents

1. Introduction

As long as it is more difficult to deploy IPv6 nodes than IPv4 nodes in today's Internet, adoption of IPv6 is going to be slow. The use of NAT in today's Internet has created certain expectations and operational conveniences, but at the cost of some important features. In particular, communications are often not really bidirectional since the device whose IP address is to be translated typically has to initiate the communication.

In order to encourage the adoption of IPv6, it is likely to be important to enable IPv6-only nodes to provide services to the existing IPv4-dominant Internet. Otherwise, if services can be provided for today's Internet only by assigning IPv4 addresses to the service-providing nodes, there is decreased economic incentive for moving to IPv6.

The approach proposed in this document should be considered a specialized form of flow management, where flows are identified by source and destination IP addresses (usually also including additional information including ports). The NAT box manages the flows, allocating and deallocating resources, and managing the traffic (albeit intrusively) according to the mutual needs of the source and destination networks.

Using the techniques proposed in this specification, communication between IPv4 nodes and IPv6 nodes can be accomplished with minimal requirements on the nodes and infrastructure:

- no dual stack
- no restriction to port-based communications
- no tunneling
- no changes to IPv4 or IPv6 hosts

Moreover, the approach is scalable because each IPv4 address used by for the incoming flows can be shared by many different IPv6-only devices. The degree of scalability is determined by the rate of arrival for new incoming connection requests, and to a lesser extent by the number of simultaneous connection requests initiated from any particular IPv4 host.

2. Terminology

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 [1].

3. Overview

Suppose that an operator wishes to support a large population of IPv6-only nodes. Also, suppose that the operator requires that the IPv6 nodes should have free access to the existing IPv4 Internet, and that customers in the existing Internet should also have free access to service-providing nodes in the IPv6-only domain, as well as to any of the other nodes in the domain for which such incoming communications would be valuable. This might, for instance, greatly simplify real-time gaming and VoIP.

Here is a proposed sequence of events. Let v6dev.foo.net be the FQDN for a v6-only node in the operator's domain. When a node in the IPv4 Internet wishes to establish a communication with v6dev, it sends a IPv4 DNS query to the name server (denoted fooNS) for foo.net. Suppose that the fooNS is programmed to supply an IPv4 address for such IPv4 DNS queries, but there is no such IPv4 address record available. Then, fooNS contacts the NAT box to get the required IPv4 address; in this context, the NAT box has the function of address allocation. Then fooNS creates a DNS reply with the appropriate A record. Importantly, fooNS does NOT store this A record for v6dev. Every distinct DNS query for v6dev could conceivably get a distinct IPv4 address allocation. The cache time for the A record is set to the minimum (either 0 or 1, depending on policy).

When fooNS sends the request to the NAT box (call it SIPNAT) for an address allocation, SIPNAT allocates the address (call it v6dev-IPv4) and creates a flow record with the following information:

- v6dev-IPv4
- the time when v6dev-IPv4 was allocated
- the IPv6 address of v6dev
- which nameserver made the request

It also sets the status of the address allocation as "pending", and sets a timeout (call it BIND_TIMEOUT) by which the allocation has to be "established".

In due time, a packet will arrive at v6dev-IPv4, which is the address of a network interface of SIPNAT. Assuming this happens before the expiration of BIND_TIMEOUT, SIPNAT "establishes" the allocation by associating the following additional information with v6dev-IPv4:

- the source address of the incoming packet (call it CNv4, for "correspondent node IPv4")

- the source port of the incoming packet
- the time of arrival
- the updated status of "established"
- a new timeout, "WAIT_TIME"

Then, SIPNAT does the address translation as detailed below and delivers the IPv6 result to v6dev.

The translation is performed as follows: Suppose the incoming IPv4 packet has:

<source IP addr, dest IP addr> == <CNv4, v6dev-IPv4>,

where v6dev-IPv4 is an address of SIPNAT.

Then the outgoing IPv6 packet gets:

<source IP addr, dest IP addr> == <IPv4-mapped addr of CNv4, v6dev>

The same rules apply for ICMP, GRE, and other protocols. Ports remain unchanged. If it is desired to use another IPv6 prefix to identify CNv4 to v6dev, the rule above can be easily modified as long as v6dev is configured appropriately. Whichever prefix is used, v6dev will use it to send packets back through SIPNAT, which then performs the reverse translation for delivery to CNv4 in the IPv4 Internet.

If, after the allocation is established, it happens that no packets flow between v6dev and CNv4, then v6dev-IPv4 is deallocated for the purpose of communications between v6dev and CNv4. v6dev-IPv4 may remain in use for other purposes. SIPNAT waits for WAIT_TIME to receive or send a packet on the (v6dev, CNv4) flow before deallocating the flow. Under the assumption that fooNS has not cached v6dev-IPv4 as the IPv4 address of v6dev.foo.net, there is no need to notify fooNS about the deallocation. If, in the future, the no-cache assumption is relaxed, a notification of the deallocation would be needed.

When the NAT box has allocated all of its available IPv4 addresses for active or pending communications, it begins to overload the available IPv4 addresses. Each IPv4 address can be allocated for use of multiple distinct communications. The same IPv4 address can be used for numerous different IPv6-only nodes, or even for multiple distinct flows to the same IPv6-only device. Each such flow is identified by source and destination IPv4 address and port numbers,

along with possibly other information to be specified. Each new IPv4 DNS query for one of the IPv6-only nodes served by SIPNAT will trigger another allocation of one of SIPNAT's IPv4 addresses. It is not clear what the maximum degree of overload should be; it will depend on the flow management performance of the IPv4 network interfaces of the NAT box.

When a new allocation (call it again v6dev-IPv4) has been made for one of SIPNAT's IPv4 addresses, incoming packets have to be inspected to determine whether they contain a new IPv4 source address, not yet associated with any other flow using v6dev-IPv4. If such a new source address is detected, the new allocation is "established", the new data is recorded, and the timeout for the new flow is set to WAIT_TIME.

In a nutshell, the incoming sessions are demultiplexed into the IPv6-only domain based on incoming IPv4 source address, not based on incoming source port number. Given the prevalence of NATs in today's Internet, the source port number takes on additional importance, because the same IPv4 address could actually be used by multiple source computers with their IP addresses hidden from the Internet. Because of this, the source port number should be used as an additional demultiplexing index. In this way, multiple instances of the same source IPv4 address could be used at the same NATv4 address as long as port numbers were available and different for the different instances of that IPv4 source address.

4. Outgoing flows, initated by an IPv6-only device.

These can be handled in any of the ways proposed, but perhaps the simple v6v4 NAT proposals are most appropriate here. Problems with v4-mapped addresses and other difficulties associated with NATs are noted in RFC 4966, but it should also be pointed out that a majority of today's Internet citizens do not seem to be overly concerned with these limitations. We should make it our first goal to make these typical users equally or more happy with IPv6, even if the NAT solution is inherently restrictive. In fact, different outgoing NAT boxes can be used for the outgoing flows, as long as the incoming flow maintains enough traffic to avoid expiration of WAIT_TIME.

5. Denial of Service

The v4-->v6 translation relies on the availability of IPv4 interfaces on the NAT box for which no new flow allocation is "pending". If a packet arrival at such a pending IPv4 interface were to cause that interface to immediately become unavailable for establishing v4-->v6 flows, there would be an easy opportunity for an attacker to mount a denial of service attack against the domain served by the source IP (SIPNAT) NAT function. Namely, the attacker could simply spray random IPv4 packets to all of the publicly accessible IPv4 network interfaces of the SIPNAT.

In order to combat this denial of service vulnerability it is necessary to avoid the loss of the pending resource. This can be done most easily by requiring pending flows to remain pending until no packets with new source IP addresses have been received at the pending address for BINDING_WAIT time. Equivalently, this means that such a pending allocation has its BINDING_WAIT timeout restarted every time a packet arrives at the IPv4 network interface with a previously unestablished source address.

A malicious attacker can still mount a denial of service attack, but it would then require a much more sustained effort. The result would be that any new pending flow allocation might collect quite a few new flow records, which would all then have to be maintained for WAIT_TIME before deallocation. But the requirement that the attacker maintain the attack for a longer time should make it easier to trace back the offending packets back to their source. Furthermore, frequently offending source IP prefixes might well be blacklisted. Packets from blacklisted prefixes could be discarded to avoid these unwanted effects.

6. Security Considerations

Any scheme which uses an allocation scheme for IPv4 addresses on the NAT box, such that the allocated resource even temporarily impacts new allocations, is vulnerable to a denial of service attack. In the case of SIPNAT, this DoS attack takes the form of flooding the DNS Request mechanism. Such malicious flooding could have the effect of depriving the IPv4 allocation for legitimate DNS Requests from legitimate correspondents.

Allocations in the pending state are vulnerable to false establishment by malicious nodes flooding packets to all of the existing IPv4 addresses of the SIPNAT box (see Section 5). There are methods to ameliorate such attacks, such as rate limiting requrests or making restrictions on the possible source IP addresses that can satisfy the flow establishment. The technique of leaving the flow pending momentarily even after a candidate packet has arrived to establish the flow, should also greatly reduce the vulnerability to this attack.

7. Acknowledgments

Thanks to Vijay Devarapalli, who provided useful ideas to make important improvements in the proposal. Thanks to Mark Andrews, who offered the solution of extending the availability of "pending" flow allocations, by restarting the BIND_TIMEOUT.

8. Normative References

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

Appendix A. Using NAT for the DNS resolution

If the NAT box is used as the authoritative name server for a special subdomain of foo.net, say for example v6only_domain.foo.net, then this design can be carried out without requiring changes to the existing DNS infrastructure. It is a matter of discussion whether or not it would be desirable to recommend the isolation of such v6-only devices and their transient A records to such subdomains.

Appendix B. Some observations about dual-stack solutions

From the standpoint of utility for conserving IPv4 address space during the transition to IPv6, dual-stack designs do not offer the advantages that are sometimes claimed.

There are three likely possibilities for a dual-stack implementation

- The IPv4 address is globally unique. This is very undesirable to make as a requirement, since then we have accomplished nothing towards the goal of making available more network-layer addresses.

- The IPv4 address is a private address, and there is a NAT box at the border of the dual-stack domain. In this case, we have NAT. Since IPv6-only hosts can work just fine with NATs, why require dual stack?

- The IPv4 address is a private address, and the dual-stack node is required to do tunnel processing on incoming v6-addressed packets that it receives. This amounts to a substantial implementation burden and, when communications occurs over a wireless medium, even more overhead.

Nevertheless, dual-stack hosts are very useful when there is a need for network nodes to offer IPv6-only applications as well as IPv4-only applications. In this scenario, the node should host a dual-stack implementation. Then, over time, as all the applications migrate to IPv6, the need for configuring the IPv4 part of the dual-stack platform will decrease until at some point the IPv4 configuration may be disregarded entirely.

Author's Address

Charles E. Perkins
WiChorus Inc.
3590 N. 1st Street, Suite 300
San Jose, CA 95134
USA

Email: charliep@computer.org