IETF Behave Working Group	C. Perkins
Internet-Draft	WiChorus Inc.
Expires: April 22, 2010	October 19, 2009

Payload-assisted Delivery for SIPNAT
draft-perkins-behave-dpinat-00.txt

Status of this Memo

Copyright Notice

Abstract

SIPNAT has been proposed as an effective method for enabling global Internet access to IPv6-only domains. New methods have been devised for accurate delivery of packets from the global Internet into the internal domain of destinations that do not share a common address space with the majority of the global Internet. These improvements can be used to augment Source-IP NAT so that perfect accuracy can be achieved in many common cases of interest.

1. Introduction

Enabling the transition from IPv4 to IPv6 will depend to a large extent upon how well businesses and online organizations can depend on IPv6 to carry on their daily operations. One way to justify such dependence on IPv6 is to assure businesses that their online services will be accessible from the entire existing IPv54 Internet.

With traditional port-mapped NAT (NAPT), this has not been possible because, for each source-destination flow, the translation parameters for the flow have had to be established by the internal network node (i.e., the node with the IP address that is incompatible with the addressing domain of the global Internet). In particular, for each such flow there needs to be an external IP address and an external port assigned. Packets arriving at the external IP address and port are then translated and retransmitted with new IP headers containing the translated IP address and port number. This works for IPv6-->IPv4 translation, IPv4-->IPv4 translation (e.g., today's Internet), and other variations as well. It is a workable solution (with various second-order difficulties) for enabling outgoing traffic to be delivered into the global Internet.

But any business requires global presence and continuous, on-demand availability. The customers have to be able to initiate contact with the business services, not the other way around. Similary for all other online service organizations (including governmental, non-profit, and family websites).

One idea for enabling such incoming translations has been proposed, called "source-IP NAT" (SIPNAT). This proposal relies on DNS to establish the required parameters for the flow translation. It has the advantage of dynamic allocation and deallocation of global IPv4 addresses for the potentially huge population of internal (say, IPv6) network nodes. This is essential for scalability. The more global IPv4 addresses, the better SIPNAT works. With as few as 128 IPv4 addresses, SIPNAT can offer reliability in excess of 99.99%, depending on the arrival rate for new flows, the cohesiveness of each flow, and other details about the statistics of the incoming traffic.

There are other protocol-specific mechanisms that can be used to assist with the translation of incoming traffic. For almost all HTTP traffic, the translation can be perfect. For SIP and peer-to-peer traffic, other mechanisms can often be employed. For some traffic, there may not be any additional mechanisms that are conveniently realizable, or even available at all. For example, "ping" and "telnet" do not make available the information needed for the protocol mechanisms in this document.

2. Failure cases for SIPNAT

SIPNAT relies on DNS Request messages to initialize a pending flow translation. The pending flow translation will become established when the first packet of the flow arrives at the external IP address on the NAT box which has been allocated for that flow. The process of establishment mainly involves inserting the source IP address of that first packet as part of the parameters for the flow translation. This also enables correct synthesis of the translated IPv6 address that will be reported to the destination, and defines the IPv4 source address for responses that are transmitted by the IPv6 destination to be translated by the source-IP NAT.

The newly allocated IP address may already be supporting several (or many) existing established flows; at any particular time, SIPNAT requires that only one pending flow translation may await establishment at the allocated IPv4 address. Because of this, SIPNAT (while effective and generally robust) does have failure modes.

*The DNS Request does not identify the actual source computer. This means the initial allocation for global Internet address on the NAT cannot be established until a packet arrives from the actual source. It is then possible for a flow translation to be assigned on a global Internet address that already is hosting a previous flow translation for the same requesting source-IP address. In other words, it is possible for a source node, which is already getting service from the NAT at a particular IPv4 address, to be accidentally allocated that same IPv4 address for a different internal destination. But, according to the rules of SIPNAT, two different IPv6 destinations for the same global Internet source cannot be translated through the same NAT IPv4 address.

*Each IPv4 NAT interface to the global Internet can sustain only one pending assignment. If too many new DNS resolutions arrive nearly simultaneously, new flow allocations may temporarily become unavailable.

*It is possible for a flow to persist even after the IPv4 address allocated for the flow has timed out. For such flows, incoming packets may be lost. To counter this, flow timeouts should be set as long as possible consistent with the target error rate. This amounts to a trade-off against the likelihood that the same source-IP address will request a new flow before all of its previous flows at the allocated NAT IP address have completed (and their flow translation parameters deactivated)

The SIPNAT document describes how to reduce or eliminate these vulnerabilities by appropriate configuration and adjustments for the timeout parameters. The first case is the most difficult case for

SIPNAT. Fortunately, according to traffic flows that have been analyzed to date, this almost never happens for small-to-medium scale websites that constitute the main use case for SIPNAT. This case does happen for very large servers, but even then it is rare, and such servers would be more efficiently handled by IVI [3] (Li, X., Bao, C., Chen, M., Zhang, H., and J. Wu, “The CERNET IVI Translation Design and Deployment for the IPv4/IPv6 Coexistence and Transition,” June 2009.).

3. Using the Payload

The basic proposal in this document is to use information contained in the payload to assist in translating the IP header from the global Internet for better-assured delivery to the proper internal host. This goes beyond previous suggestions for careful configuration and management of the NAT interface.

For instance, almost all HTTP GET traffic has the host destination host name as part of the HTTP payload:

> http.host: news.google.com
>
> http.host: my-IPv6-server.example-operator.com

For all such incoming traffic, translation can be performed with 100% accuracy. Current experience with border routers offering DPI features indicates that the translation can be done at wire speed. It is expected that this one simple observation will, for all practical purposes, eliminate any remaining ambiguities about the delivery of HTTP traffic.

Even in the unlikely event that the "http.host" field is not present in the HTTP GET traffic, web traffic offers other possibilities for guiding correct traffic. For example, the pathname of the object webpage is typically unique to the destination. In other words, it is rarely the case that two different destinations in the same domain would use the same pathname to identify any webpages hosted by those destinations. If a database of associations between pathnames and actual destinations is maintained, any such traffic containing the pathname for a destination can be delivered correctly. Information about rare cases where duplicate pathnames occur can also be maintained. Even if the pathname can narrow down the selection to a choice between a few competing websites, the other context for the flow translation will typically be sufficient for accurate delivery.

Other protocols, such as SIP, also typically utilize URIs and URLS that provide domain names identifying the desired destination. For each such protocol, the DPI methods required for extracting the destination information will be different, but the principle is the same.

4. Peer-to-Peer

After HTTP traffic, the next major source of traffic on the Internet is Peer-to-Peer traffic. This is typically filesharing traffic, often using the BitTorrent protocol. In order to understand the interactions of such traffic with SIPNAT, we will use BitTorrent as an example. BitTorrent relies on four different kinds of protocol entities.

*Directory of trackers

*Trackers

*Servers

*Clients

A client uses some method for finding a tracker that maintains information about servers offering a particular file of interest. Once contact with a tracker is established, the client obtains a list of file segments, and for each file segment, a list of servers that offer availability for that file segment. A client can select one or more servers for each segment of the desired file. If a transfer for a specific file segment from one of the selected servers does not complete, the client can pick another server for that file segment. Eventually, all segments will be collected together for assembly into the complete file of interest.

With this as context, it should be considered which configurations are of most interest. For example, consider an IPv6-only server offering segments to clients on the global Internet. In many cases, this sort of service will work just fine, especially if the clients attempt to resolve the server's domain name before issuing the request for the file segment. However, the tracker often supplies the IP address of the server instead of the server's domain name. For such communications, the client would expect to make contact with the server without any intermediate DNS resolution to obtain the IP address of the server. In this case, the SIPNAT would not have the opportunity to allocate the necessary IPv4 address for the flow translation.

Nevertheless, it is still possible to handle such incoming traffic, based on detailed methods for inspecting peer-to-peer traffic payloads. This is to be specified in a companion ALG document describing mechanisms for handling IPv4-->IPv6 translation for the lists of servers provided by trackers. One simple solution is just to set up IVI-style network interfaces for the servers. For dynamic allocations of IPv4 network interfaces on the NAT router, additional payload inspection and alterations are needed.

5. Other traffic

Aside from HTTP, peer-to-peer, and SIP traffic, other protocol services should be considered on a case-by-case basis.

For instance, DNS traffic is extremely common on the Internet. However, it is most likely not a suitable candidate for such protocol translations as typically handled by NATs. Therefore, for the purposes of this document, we may consider DNS traffic to be out of scope for the translation problem.

Mobile IP signaling is not a good candidate for such protocol translations, because a client is either a Mobile IPv4 mobile node or a Mobile IPv6 mobile node, and there are already methods specified by which a Mobile IPv6 mobile node can access its home agent by way of IPv4 addresses.

It needs to be determined whether or not mail servers that have IPv6 addresses only should be accessible by way of SIPNAT. It seems more likely, even if they should be accessible at all to the existing IPv4 global Internet, they would be either dual-stack already, or else good candidates for assignment of a permanent (IVI-style) IPv4 address on the NAT.

FTP does not seem to offer a good way to identify the destination by way of the payload-oriented mechanisms described earlier in this document. Such FTP services are better handled by way of IVI-style static address assignment. For most of the cases of interest, file access is more likely mediated by HTTP access instead of FTP, and the remaining cases are typically those more appropriate for static assignment anyway.

Telnet is not used very often any more, and anyway is likely to be handled very well by SIPNAT except for cases when the client attempts to contact a destination by using the raw IP address. SIPNAT does not offer a solution for that case.

IKEv2 [2] (Kaufman, C., “Internet Key Exchange (IKEv2) Protocol,” December 2005.) has been designed to work across NAT boundaries. Consequently, it does not require the use of IP addresses as part of the IKEv2 message payload. In the case of SIPNAT, the Notify payloads NAT_DETECTION_SOURCE_IP and NAT_DETECTION_DESTINATION_IP will correctly indicate the presence of address translation. If use of a well-known IPv6 prefix is used to translate the IPv4 address, there may be an opportunity to precisely identify the type of NAT as "SIPNAT".

... think about SSH ...

6. Segregation by access type

HTTP is a particularly common case that is easy for SIPNAT to handle with the extensions described in this document. For destinations that only offer services by way of HTTP, all traffic can be delivered accurately based on the payload. This means that many different allocations could be made at the same time without danger of ambiguity. Effectively, the restriction for only one pending allocation at a time could be removed.

Suppose, then, that there is a special class of destinations that only offer HTTP service. Also suppose that the NAT is equipped to detect the "http.host" field of incoming HTTP GET requests. For every destination in this special class, each DNS Request could return the same IPv4 address in the DNS Reply. The same sort of flow translation would be set up for each HTTP destination assigned to the IPv4 address, but the destination IP address parameter would be determined by the payload, and matched against the initial allocation as determined by the DNS entity. However, packet delivery should still be granted only for destinations that had been allocated to the IPv4 address by way of a preceding DNS request. This eligibility requirement should be a matter of operator policy. For such traffic, the flow timeout parameter could be configured to a much larger value.

This scheme could potentially be extended to allow HTTP access to some destinations identified by IP address (not hostname) in URLs. In other words, the pathname is located at a raw IP address instead of the usual domain name for the destination node. For these cases, the payload can be delivered accurately, but no DNS Request would be received to trigger the allocation of a pending flow translation.

Such unusual URLs can be supported under the typical configuration choice for HTTP servers as has been described. Again, it is a matter for operator choice whether it should be appropriate to provide that support. Moreover, it is not clear how a source would ever get such a URL with an IPv4 address to represent the IPv6-only destination.

7. References

7.1. Normative References

[1] Perkins, C., “Translating IPv4 to IPv6 based on source IPv4 address,” draft-perkins-sourceipnat-00 (work in progress), October 2009 (TXT).

7.2. Informative References

[2] Kaufman, C., “Internet Key Exchange (IKEv2) Protocol,” RFC 4306, December 2005 (TXT).

[3]

	Li, X., Bao, C., Chen, M., Zhang, H., and J. Wu, “The CERNET IVI Translation Design and Deployment for the IPv4/IPv6 Coexistence and Transition,” draft-xli-behave-ivi-02 (work in progress), June 2009 (TXT).
[4]	Bagnulo, M., Sullivan, A., Matthews, P., and I. Beijnum, “DNS64: DNS extensions for Network Address Translation from IPv6 Clients to IPv4 Servers,” draft-ietf-behave-dns64-00 (work in progress), July 2009 (TXT).

Author's Address

Charles E. Perkins
WiChorus Inc.
3590 N. 1st Street, Suite 300
San Jose CA 95134
USA
Email: charliep@computer.org