Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 14, 2008

M. Bagnulo
Huawei Labs at UC3M

November 11, 2007

IPv6 - IPv4 Translators (NAT64) - Problem Statement and Analysis
draft-bagnulo-v6ops-6man-nat64-pb-statement-00

Status of this Memo

Copyright Notice

Abstract

RFC 4966 published on July 2007 deprecates the NAT-PT tool, the mechanism defined by the IETF to enable communications between IPv4 only nodes with IPv6 only nodes, letting the dual-stack approach as the preferred mechanism to enable nodes to be able to communicate with v6 and v4 nodes. However, there are several reasons why the dual stack approach may not be adequate for a number of scenarios. For once, the dual-stack approach imposes the management of two networks in a site, the v6 one and the v4 one, increasing the costs of using IPv6. In addition, as IPv4 public address space is depleted, it will no longer possible to access to IPv4 public addresses, making dual stack nodes even less attractive. It is then considered necessary to explore alternative mechanisms that enable communication of v6 nodes and v4 nodes, other than the deprecated NAT-PT. In order to do that, the first that is needed is to understand what capabilities are required and what constrains affect the design space. The goal of this document is to state the different capabilities that can be required to IPv4 - IPv6 translators (hereafter called NAT64) and the constraints that affect the solution space.

Table of Contents

1. Introduction

RFC 4966 published on July 2007 deprecates the NAT-PT tool, the mechanism defined by the IETF to enable communications between IPv4 only nodes with IPv6 only nodes, letting the dual-stack approach as the preferred mechanism to enable nodes to be able to communicate with v6 and v4 nodes. However, there are several reasons why the dual stack approach may not be adequate for a number of scenarios. For once, the dual-stack approach imposes the management of two networks in a site, the v6 one and the v4 one, increasing the costs of using IPv6. In addition, as IPv4 public address space is depleted, it will no longer possible to access to IPv4 public addresses, making dual stack nodes even less attractive. It is then considered necessary to explore alternative mechanisms that enable communication of v6 nodes and v4 nodes, other than the deprecated NAT-PT. In order to do that, the first that is needed is to understand what capabilities are required and what constrains affect the design space. The goal of this document is to state the different capabilities that can be required to IPv4 - IPv6 translators (hereafter called NAT64) and the constraints that affect the solution space.

2. Scenario

The scenario of operation of NAT64-type of mechanisms is the following. We have a IPv4-only node in the v4 Internet and an IPv6-only node in the v6 Internet. The goal is to enable communications between these two nodes. In order to do that, we install a NAT64 box with two interfaces, one in the v4 Internet and the other one in the v6 Internet. We assume that there is v6 communication between the IPv6-only node and the v6 interface of the NAT64 and that there is v4 connectivity between the IPv4-only node and the v4 interface of the NAT64 box. The type of v4 addressing that is used in the v4-only node and in the v4 interface of the NAT64 is part of the design choices that are analyzed below.

```
    +----------------------------------+-------------------------------+
    |           v4 Internet            |         v6 Internet           |
    | +-------+                  +------------+               +-------+ |
    | | IPv4  |                  |   NAT64    |               | IPv6  | |
    | | only  |------------------|  v4 |  v6 |---------------| only  | |
    | | node  |                  | Translator |               | node  | |
    | +-------+                  +------------+               +-------+ |
    |                                  |                               |
    +----------------------------------+-------------------------------+
```

3. Supported application behavior

The general purpose of NAT64 type of mechanisms is to enable communication between a v4-only node and a v6-only node. However, there is wide range of type of communication, when considering how they handle the IP addresses. So, in order to properly characterize the problem in hand, we need to do a more depth analysis of the different application behavior in terms of the usage of their IP addresses. We will next present a taxonomy of the behavior of the application with respect of how they use the IP address and the support of the different type of behavior will impose a different set of constraints to the design of a NAT64 mechanisms. It is then important to decide which type of application behavior will be supported before starting to design a NAT64 mechanism. The proposed taxonomy is heavily based on the one presented in section 1.1 of draft-ietf-shim6-app-refer-00.txt.

The proposed application behavior taxonomy is the following:

Short-lived local handle. The IP addresses is never retained by the application. The only usage is for the application to pass it from the DNS APIs (e.g., getaddrinfo()) and the API to the protocol stack (e.g., connect() or sendto()). This type of communication can be either initiated by the v4-only node or by the v6-only node, resulting in two type of behaviors, v4-initiated short lived local handle and v6-initiated short lived local handle.

Long-lived application associations. The IP address is retained by the application for several instances of communication. However, it is always the same node that initiates the communication. This type of communication can be either initiated by the v4-only node or by the v6-only node, resulting in two type of behaviors, v4-initiated long-lived associations and v6-initiated long-lived associations.

Callbacks. The application at one end retrieves the IP address of the peer and uses that to later communicate "back" to the peer. This type of communication can be either initiated by the v4-only node or by the v6-only node, resulting in two type of behaviors, v4-initiated callback, meaning that the initial communication is initiated by the v6-only node, and later the v4-only node initiates the callback, and v6-initiated callback, meaning that the initial communication is initiated by the v4-only node, and later the v6-only node initiates the callback .

Referrals. In an application with more than two parties, party B takes the IP address of party A and passes that to party C. After this party C uses the IP address to communicate with A. In this type of communication, the following 6 sub-cases are possible.

- A and B are v6-only nodes and C is a v4-only node;
- A and C are v6-only nodes and B is a v4-only node,
- B and C are v6-only nodes and A is a v4-only node,
- A and B are v4-only nodes and C is a v6-only node;
- A and C are v4-only nodes and B is a v6-only node,
- B and C are v4-only nodes and A is a v6-only node,

"Identity" comparison. Some applications might retain the IP address, not as a means to initiate communication as in the above cases, but as a means to compare whether a peer is the same as another peer. While this is insecure in general, it might be something which is used e.g., when TLS is used. This type of communication results in two sub-cases, when the v4-only node performs comparison of the v6-only node identity, and when the v6-only node performs comparison of the v4-only node identity

Discussion: is there another type of application that embed IP addresses in the application data that doesn't fit in the previous cases?

4. Placement of the NAT64 mechanisms

Another aspect that is critical to design a NAT64 mechanism is the placement of the mechanisms involved. In other words, what elements can be modified/updated to support the NAT64 mechanisms. We assume that the NAT64 box supports a set of mechanisms that are the core part of the solution, but some approaches may require the modification of additional elements. In particular, we can identify the following additional elements that may require modification to support a NAT64 approach.

Modification to v4-only nodes: one option is to require modification to existent v4-only nodes in order to support the NAT64 mechanism. This option would impose high deployment costs, because the existent base of v4-only nodes is really big and there is no incentives for the v4-only nodes to install such mechanism, since it seems unlikely that v4-only nodes will have a strong need to communicate with v6-only nodes (at least at the initial stages of v6 deployment). However, it may be possible that this is the only viable solution for supporting some type of application behavior.

Modification to v6-only nodes: Another option is to require modifications to v6-only nodes. This option seems much more acceptable, since the existent base of v6-nodes is relatively small and there would be a strong incentive for v6-only nodes to communicate with v4-only nodes, since most of the contents are available only in v4 today. However, imposing modifications to v6-only nodes does make deployment of the solution more difficult, since update of current v6-implementations is needed. In addition, there is an architectural consideration, that we would be imposing v6-only nodes to support "NAT hacks" in order to enable communication with the v4 world, and that those modifications may stay forever, even when the need for communication with the v4-Internet is not so pressing.

Modification to both v4-only nodes and v6-only nodes. Another option is to require updates to both v4-only nodes and also to v6-only nodes. Needless to say that this would be the option with higher deployment costs.

No modification. Another option is that the NAT64 mechanisms does not require modification to any host and that the mechanism is fully contained in the NAT64 box. This was the case of the previously defined NAT-PT approach. However, it may be challenging to design a solution with this constraint that does not suffer the limitations suffered by the NAT-PT mechanism that lead the IETF community to deprecate it.

Another consideration related to the modification imposed by a NAT64 approach is about what elements in the nodes need to be updated. In particular, it is important to determine if only the IP layer on the affected nodes needs to be modified or f other elements in the nodes needs to be updated. In particular, it is critical to determine if applications need to e modified in order to support the NAT64 mechanism.

5. v4 addressing consideration

We assume that both the v6-only nodes and the v6 interface of the NAT64 boxes will have routable IPv6 addresses. However, on the v4 side, there are more options. Either the v4 interface of the NAT64 boxes and/or the v4-only nodes can have either v4 private addresses or v4 public addresses. Actually, it is possible that the different combinations make sense. It seems clear that the case where public v4 addresses are used in both the v4 interface of the NAT64 box and the v4-only nodes is relevant. The case where the v4-only node has a private v4 address and the NAT64 box has a public address seems also possible, but here it seems reasonable to assume that a NAT box will exist between the v4 only node and the NAT64 box. The case where both the v4 node and the NAT64 box have v4 private addresses could also make sense, since this could apply to a scenario where a site that has v4 private addresses and v6 addresses could try to use a NAT64 box internally. The last case, where the v4 node has public address and the NAT64 box has a private address seems harder to

justify though.

Another consideration related to v4 addressing of the NAT64 approach is the number of addresses required by the NAT64 box. It is possible that some NAT64 approaches require a pool of v4 addresses instead of a single v4 address. Considering the status of the v4 address space consumption, it may not be feasible to use a NAT64 approach that require a big number of v4 public addresses.

6. Name-space considerations

One of the major choices that are faced when designing a NAT4 mechanism that enable communication initiated by the v4-only node towards a v6-only node. In this case, the v4 only node needs to identify the v6 only node and the problem is that there is no mean to permanently map the v6 address space in the v4 address space. So in order to enable a v4-only node to identify a v6-only node a name space other than the IPv4 address space is needed. We will next discuss some options that could be considered to identify v6 nodes in the v4 world.

A first option is to use IPv4 addresses to identify IPv6 nodes. The problem is that the v6 address space is much bigger than the v4 address space, so it is not possible to do permanent mapping between these two. This basically implies that dynamic mapping between a given v4 address and different v6 addresses are established. While this works for some type of application behavior, it does not support others, such as communications initiated by a v4 node towards a v6 node in a general case (it is possible for a given subset of v6 nodes, but not as a general solution)

A second option is to use IPv6 addresses themselves. In this case, the IPv4 node is aware of the IPv6 address of the destination and it uses it to identify the target at the NAT64 box. This option would likely imply modifications in the v4 nodes.

A third option is to use FQDN to identify nodes. In this case v4 nodes identify v6 nodes using FQDNs, which is already supported in the v4 world. The difficulties with such a approach is that DNS ALG are likely to be required.

A fourth option is to use a combination of IPv4 address, transport protocol and port for identification of a v6 node or a v6 flow.

7. Other considerations

7.1. Transparency considerations

Another design option that need to be addressed is whether the NAT64 box will be transparent or not to the end nodes. One option is that the nodes are aware of the NAT64 box and they establish a communication with it to configure their translation. Th other option is that the NAT64 box is transparent and that the end nodes just establish the communication (even if they know they are communicating with a node that supports a different version of IP, there is no explicit communication between the end nodes and the NAT64 box).

8. Security considerations

TBD

Author's Address

Marcelo Bagnulo
Huawei Labs at Universidad Carlos III de Madrid
Av. Universidad 30
Leganes, Madrid 28911
SPAIN

Phone: 34 91 6249500
Email: marcelo@it.uc3m.es
URI: http://www.it.uc3m.es

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).