Network Working Group
Internet-Draft
Intended status: Informational
Expires: March 25, 2015

Y. Bai
C. Bao
CERNET Center/Tsinghua University
K. Yin
Cisco Systems
X. Li
CERNET Center/Tsinghua University

September 21, 2014

IPv4/IPv6 Transition Practice in OpenStack
draft-ybai-ipv4v6-transition-practice-in-openstack-00

Abstract

OpenStack is a free and open-source software cloud computing platform. It is primarily deployed as an infrastructure as a service (IaaS) solution. However, OpenStack is designed mainly for IPv4, it internally uses [RFC1918] addresses and heavily relies on NAT to map RFC1918 addresses to public IPv4 addresses known as floating IP addresses for the external access. Due to the different nature of IPv6 and IPv4, the IPv6 support for the OpenStack is still in the early stage. In this document, two mechanisms are presented to provide IPv4/IPv6 dual stack external access for the OpenStack, one scenario is internal IPv4 and uses stateful IPv4/IPv6 translator for the external IPv6 access, and another scenario is internal IPv6 and uses stateless IPv4/IPv6 translation for the external IPv4 access. Both mechanisms have been deployed in CERNET and providing services to the global IPv4/IPv6 Internet.

Table of Contents

1. Introduction

The concept of cloud is growing rapidly, and open-source cloud platforms such as OpenStack are more and more popular. The network models inside OpenStack cloud requires public IPv4 addresses for external access. It's foreseeable that the exhaustion of IPv4 global addresses would be one of the bottlenecks on deploying clouds in the future. While the important and urgency of IPv4/IPv6 transition has been studied widely, the support of IPv6 in OpenStack is still in its early stage. For example, the private addresses assigned to instances are not favored in IPv6, and the concepts like "Floating IP" have no counterparts in IPv6. Therefore, the structure of OpenStack should be extended to meet the need of IPv6 clouds. This

document presents the analysis to extend OpenStack to IPv6. Based on the extensions presented in this document, it can be shown that using stateful IPv4/IPv6 translator, the external IPv6-only hosts can access the existing IPv4-only VMs in OpenStack. We have installed a new module in OpenStack which enables internal IPv6 support for OpenStack. By using stateless IPv4/IPv6 translator, external IPv4-only hosts can access the IPv6-only VMs in OpenStack.

2. IPv4/IPv6 access to IPv4-only Cloud

2.1. Current IPv4 OpenStack Network Structures

Current IPv4 OpenStack network structure can be described as follows. Rather than directly connect to public network, VM instances are resided in "Private networks" under their respective tenants and are assigned with private addresses. To be accessed from the external networks, those private networks should be linked to public networks via a virtual device called "virtual router". Some key concepts and typical techniques of this structure are:

VLAN: VLAN are used to accomplish the segregation of tenants, each tenant receives one VLAN tag.

Subnets: Each tenant can further divide their networks into "subnets" with different IP address pool.

NAT: When instances need to access external networks, they share a public address that is owned by the virtual router.

Floating IP: This is the core concept of OpenStack network structure. When instances need to be accessed from external networks, each instance associates its private address with a public address. The private address is mapped to the public address on outgoing flow and the public address is mapped to the private address on ingoing flow, and the mapping is implemented by the virtual router.

2.2. IPv4 Accessibility directly

In this scenario, instances without Floating IPs could access external IPv4 Internet via NAT as shown in Figure 1. Here, 40.40.40.40 is the address of the gateway of the virtual router, and 30.30.30.30 is the server that instance want to access. Port1 and port2 are two different ports, the dynamic mapping relationship is maintained by the virtual router.

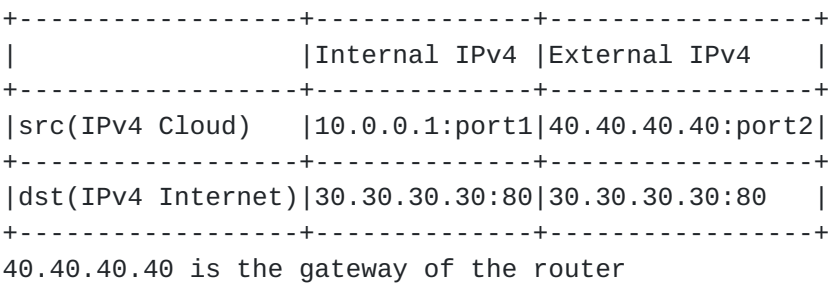

Figure 1: IPv4 cloud access the IPv4 Internet

With Floating IP, IPv4 Internet can access the VM instances as shown in Figure 2. Here, address 10.0.0.1 is associated with address 40.40.40.41, their static mapping is implemented in the virtual router.

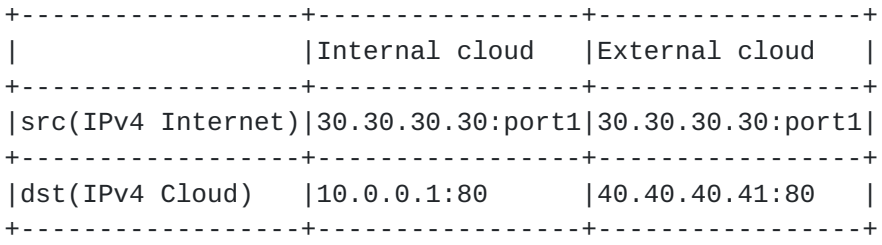

Figure 2: IPv4 Internet access IPv4 cloud

2.3. IPv6 Accessibility via IPv4/IPv6 translator

This scenario corresponds to the Scenario 3 defined in [RFC6144]: IPv6 Internet accesses IPv4 network, where the cloud is considered as an IPv4 network.

As described in [RFC6052], [RFC6145], [RFC6146] and [RFC6147], the IPv6 prefix, an IPv4 pool to represent the external IPv6 hosts, the pool for the floating IP, and the DNS AAAA record to represent IPv4-converted floating IP are configured. The example is shown in Figure 3, where the IPv6 prefix=2001:da8:e164::/48, the IPv4 pool to represent the external IPv6 hosts=202.38.97.0/24, the pool for the floating IP=121.194.167.196/24.

	External IPv6	Xlate IPv6 side
src(IPv6 Internet) port	2001:250:3:0:7a26:cbff:: random port	2001:da8:e164:ca26:6118:: random port
dst(IPv4 Cloud) port	2001:da8:e164:79c2:a7c4:: 80	2001:da8:e164:79c2:a7c4:: 80

(cont.)

	Xlate IPv4 side	Internal IPv4
src(IPv6 Internet) port	202.38.97.24 random port	202.38.97.24 random port
dst(IPv4 Cloud) port	121.194.167.196 80	10.10.1.5 80

Figure 3: IPv6 Internet accesses IPv4 Cloud

In this scenario, IPv6 Internet could gain the ability to access those IPv4 clouds, as long as the instances are associated with Floating IPs.

3. IPv4/IPv6 access to IPv6-only Cloud

3.1. Analysis and recommendations for the Internal Structure of IPv6 OpenStack

Since the internal support for IPv6 on OpenStack is still in its very early stage, the IPv6 extension must be developed inside the OpenStack cloud. The building blocks for the extension includes the IPv6 address assignment and the floating IPv4 equivalent mechanisms. For address assignment, several different mechanisms like DHCPv6 and SLAAC could be used. For external IPv6 access, three possible solutions could be used, they're NAT66 (like NPT66 defined in [RFC6296]), ND Proxy (defined in [RFC4389]) and enabling autoconfiguration of the IPv6 routing protocols (OSPF, BGP).

For NAT66, private addresses like ULA could be used in internal network, while they're associated to a public address, and this structure is similar to the IPv4 structure. However, as NAT is deprecated in IPv6 to ensure end-to-end transparency, this scheme is strongly opposed by IPv6 community.

For ND Proxy, based on the hierarchical address structure, the proxy rules can be set at the edge of the cloud. Therefore, the router interface will take the responsibility to respond all the neighbour discovery request for internal networks. At the current stage, it is considered that this structure requires minimum changes to both the OpenStack internal structure, as well as the IPv6 architecture.

For enabling and autoconfiguration of the IPv6 routing protocols (OSPF, BGP), the virtual router of OpenStack may interact with the external routers to exchange the routing information to create routes. However, the function of virtual router in OpenStack is based on Linux Kernel, and support for a complicated router protocol would be overkilling for those virtual routers.

Therefore, it is clear that:

1. The NAT66 is almost identical to the current OpenStack IPv4 network structure, while the ND proxy and routing protocol are not.

2. The NAT66 doesn't maintain the end-to-end transparency in IPv6, while ND proxy and routing protocol do.

3. The NAT66 scheme will use ULA address inside the cloud, while ND proxy and routing protocol are using global IPv6 addresses.

4. For ND Proxy, any global address with prefix longer than /64 could be used and therefore the SLAAC should not be used.

5. Routing protocol needs interation with the upstream router, while NAT66 and ND proxy do not.

As a conclusion, we recommend ND Proxy as the best solution among the three mentioned above, as it requires the minimum changes to both internal and external networks, and no additional configurations are required on upstream router.

3.2. IPv4 Accessibility via Translation

This scenario corresponds to Scenario 2 defined in [RFC6144], where IPv6 Internet can access IPv6 cloud directly, and IPv4 Internet can access IPv6 cloud using stateless translator. Using the ND proxy mechanism described in Section 3.1, the IPv6 Internet Access IPv6 Cloud and the IPv4 Internet Access IPv6 Cloud are shown in Figure 4 and Figure 5, respectively.

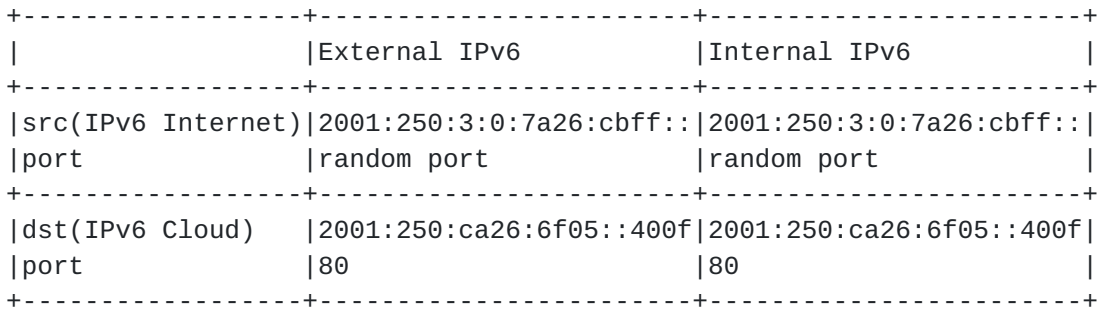

Figure 4: IPv6 Internet accesses IPv6 Cloud

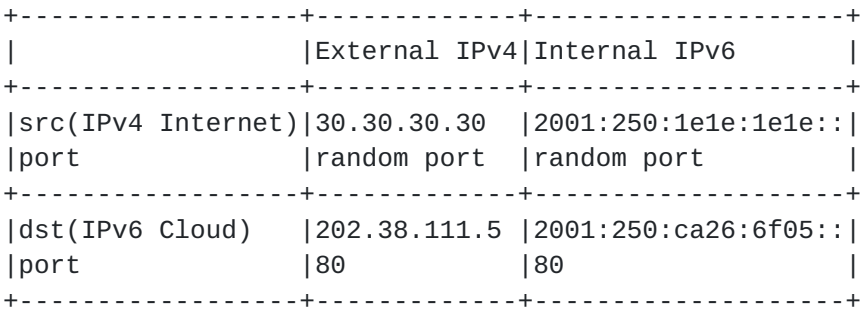

Figure 5: IPv4 Internet accesses IPv6 Cloud

3.2.1. 1:N Stateless Translation

Due to the IPv4 address depletion, the public IPv4 addresses need to be shared for the cloud environment. Using the method described in [I-D.bcx-address-fmt-extension], the IPv4 address sharing ratio can be achieved as high as 4096. The example is shown in Figure 6.

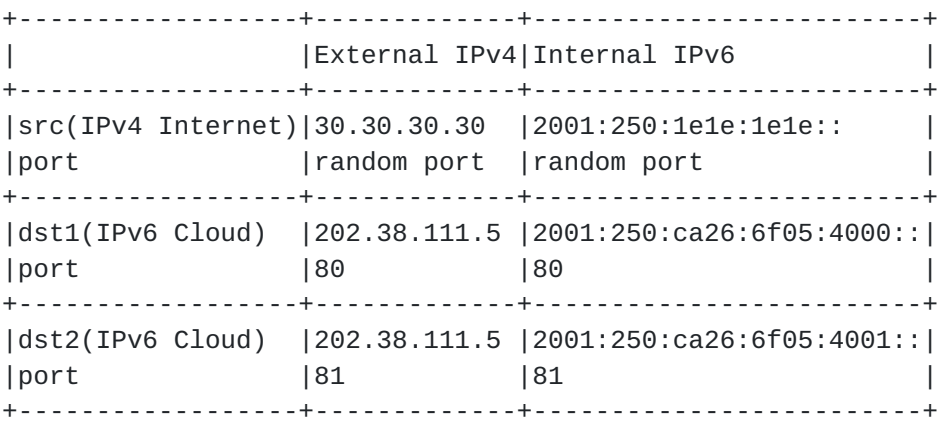

Figure 6: IPv4 Internet Access IPv6 Cloud with IPv4 address sharing

3.2.2. HTTP Redirection for Web Servers

However, this address format limits the use of port in instances, for example, the suffix of the port may be fixed to the offset defined in the address. For a certain server, rather than use the standard port numbers like 80 for HTTP server, the server must use non standard ports like 81 or 82. To solve this problem, redirection could be used for web servers. For web server, translator would look up the domain name, then redirect to the corresponding VM instance. In this way, different cloud servers could share the same IPv4 address and the same (standard) port to provide service. An example is shown in Figure 7.

	External IPv4	Internal IPv6
src(IPv4 Internet) port	30.30.30.30 random port	2001:250:1e1e:1e1e:: random port
dst1(IPv6 Cloud) port domain name	202.38.111.5 80 vm1.example.com	2001:250:ca26:6f05:4001:: 80 vm1.example.com
dst2(IPv6 Cloud) port domain name	202.38.111.5 80 vm2.example.com	2001:250:ca26:6f05:4002:: 80 vm2.example.com

dst1 and dst2 share the common IPv4 global addresses 202.38.111.5 and standard port 80, they provide services by standard port 80.

Figure 7: IPv4 Internet Access IPv6 Cloud with HTTP redirection

4. Summary

In current IPv4-only OpenStack, by using extended translation mechanisms, IPv6 Internet could access IPv4 cloud with little modification inside the cloud, while native IPv4 accessibility is remained. By extending IPv6 support in OpenStack, including address assignment mechanisms and ND Proxy, IPv6 in OpenStack cloud is enabled. By deploying improved translators and proxies, the IPv6-only cloud can provide services like SSH (in the case of IPv4 address sharing, using address plus port) and HTTP (in the case of IPv4 address sharing, using address plus port directly or DNS with HTTP redirect) to both native IPv6 and IPv4 with IPv4 address sharing ability.

5. Security Considerations

This document does not introduce any new security considerations.

6. IANA Considerations

None.

7. Acknowledgments

The authors would like to acknowledge the following contributors of this document: Rong Jin, Qiuhan Ding and Weicai Wang.

8. Normative References

[I-D.bcx-address-fmt-extension]
Bao, C. and X. Li, "Extended IPv6 Addressing for Encoding Port Range", draft-bcx-address-fmt-extension-02 (work in progress), October 2011.

[RFC1918] Rekhter, Y., Moskowitz, R., Karrenberg, D., Groot, G., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, February 1996.

[RFC4389] Thaler, D., Talwar, M., and C. Patel, "Neighbor Discovery Proxies (ND Proxy)", RFC 4389, April 2006.

[RFC6052] Bao, C., Huitema, C., Bagnulo, M., Boucadair, M., and X. Li, "IPv6 Addressing of IPv4/IPv6 Translators", RFC 6052, October 2010.

[RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, April 2011.

[RFC6145] Li, X., Bao, C., and F. Baker, "IP/ICMP Translation Algorithm", RFC 6145, April 2011.

[RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, April 2011.

[RFC6147] Bagnulo, M., Sullivan, A., Matthews, P., and I. van Beijnum, "DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers", RFC 6147, April 2011.

[RFC6296] Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", RFC 6296, June 2011.

Authors' Addresses

Yi Bai
CERNET Center/Tsinghua University
Room 225, Main Building, Tsinghua University
Beijing 100084
CN

Phone: +86 10-62785983
Email: yibai.thu@gmail.com

Congxiao Bao
CERNET Center/Tsinghua University
Room 225, Main Building, Tsinghua University
Beijing 100084
CN

Phone: +86 10-62785983
Email: congxiao@cernet.edu.cn

Kevin Yin
Cisco Systems
No. 2 Jianguomenwai Ave, Chaoyang District
Beijing 100022
China

Phone: +86-10-8515-5094
Email: wkyin@cisco.com

Xing Li
CERNET Center/Tsinghua University
Room 225, Main Building, Tsinghua University
Beijing 100084
CN

Phone: +86 10-62785983
Email: xing@cernet.edu.cn