Network Working Group
Internet-Draft
Intended status: Best Current Practice
Expires: February 10, 2019

B. Sarikaya
Denpel Informatique
L. Dunbar
Huawei USA
B. Khasnabish
ZTE (TX) Inc.
T. Herbert
Quantonium
S. Dikshit
Cisco Systems
August 9, 2018

Virtual Machine Mobility Protocol for L2 and L3 Overlay Networks
draft-ietf-nvo3-vmm-04.txt

Abstract

This document describes a virtual machine mobility protocol commonly used in data centers built with overlay-based network virtualization approach. For layer 2, it is based on using a Network Virtualization Authority (NVA)-Network Virtualization Edge (NVE) protocol to update Address Resolution Protocol (ARP) table or neighbor cache entries at the NVA and the source NVEs tunneling in-flight packets to the destination NVE after the virtual machine moves from source NVE to the destination NVE. For Layer 3, it is based on address and connection migration after the move.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on February 10, 2019.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

Data center networks are being increasingly used by telecom operators as well as by enterprises. In this document we are interested in overlay-based data center networks supporting multitenancy. These networks are organized as one large Layer 2 network geographically distributed in several buildings. In some cases geographical distribution can span across Layer 2 boundaries. In that case need arises for connectivity between Layer 2 boundaries which can be achieved by the network virtualization edge (NVE) functioning as

Layer 3 gateway routing across bridging domain such as in Warehouse Scale Computers (WSC).

Virtualization which is being used in almost all of today's data centers enables many virtual machines to run on a single physical computer or compute server. Virtual machines (VM) need hypervisor running on the physical compute server to provide them shared processor/memory/storage. Network connectivity is provided by the network virtualization edge (NVE) [RFC8014]. Being able to move VMs dynamically, or live migration, from one server to another allows for dynamic load balancing or work distribution and thus it is a highly desirable feature [RFC7364].

There are many challenges and requirements related to migration, mobility, and interconnection of Virtual Machines (VMs)and Virtual Network Elements (VNEs). Retaining IP addresses after a move is a key requirement [RFC7364]. Such a requirement is needed in order to maintain existing transport connections.

In L3 based data networks, retaining IP addresses after a move is simply not possible. This introduces complexity in IP address management and as a result transport connections need to be reestablished.

In view of many virtual machine mobility schemes that exist today, there is a desire to define a standard control plane protocol for virtual machine mobility. The protocol should be based on IPv4 or IPv6. In this document we specify such a protocol for Layer 2 and Layer 3 data networks.

2. Conventions and 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 [RFC2119] and [RFC8014].

This document uses the terminology defined in [RFC7364]. In addition we make the following definitions:

Tasks. Tasks are the generalization of virtual machines. Tasks in containers that can be migrated correspond to the virtual machines that can be migrated. We use task and virtual machine interchangeably in this document.

Hot VM Mobility. A given VM could be moved from one server to another in running state.

Warm VM Mobility. In case of warm VM mobility, the VM states are mirrored to the secondary server (or domain) at a predefined (configurable) regular intervals. This reduces the overheads and complexity but this may also lead to a situation when both servers may not contain the exact same data (state information)

Cold VM Mobility. A given VM could be moved from one server to another in stopped or suspended state.

Source NVE refers to the old NVE where packets were forwarded to before migration.

Destination NVE refers to the new NVE after migration.

Packets in flight refers to the packets received by the source NVE sent by the correspondents that have old ARP or neighbor cache entry before VM or task migration.

Users of VMs in diskless systems or systems not using configuration files are called end user clients.

3. Requirements

This section states requirements on data center network virtual machine mobility.

Data center network SHOULD support virtual machine mobility in IPv6.

IPv4 SHOULD also be supported in virtual machine mobility.

Virtual machine mobility protocol MAY support host routes to accomplish virtualization.

Virtual machine mobility protocol SHOULD not support triangular routing except for handling packets in flight.

Virtual machine mobility protocol SHOULD not need to use tunneling except for handling packets in flight.

4. Overview of the protocol

Layer 2 and Layer 3 protocols are described next. In the following sections, we examine more advanced features.

4.1. VM Migration

Being able to move Virtual Machines dynamically, from one server to another allows for dynamic load balancing or work distribution and thus it is a highly desirable feature. In a Layer-2 based data center approach, virtual machine moving to another server does not change its IP address. Because of this an IP based virtual machine mobility protocol is not needed. However, when a virtual machine moves, NVEs need to change their caches associating VM Layer 2 or Medium Access Control (MAC) address with NVE's IP address. Such a change enables NVE to send outgoing MAC frames addressed to the virtual machine. VM movement across Layer 3 boundaries is not typical but the same solution applies if the VM moves in the same link such as in WSCs.

Virtual machine moves from its source NVE to a new, destination NVE. After the move the virtual machine IP address(es) do not change but this virtual machine is now under a new NVE, previously communicating NVEs will continue to send their packets to the source NVE. Address Resolution Protocol (ARP) cache in IPv4 [RFC0826] or neighbor cache in IPv6 [RFC4861] in the NVEs need to be updated.

It may take some time to refresh ARP/ND cache when a VM is moved to a new destination NVE. During this period, a tunnel is needed so that source NVE forwards packets to the destination NVE.

In IPv4, the virtual machine immediately after the move should send a gratuitous ARP request message containing its IPv4 and Layer 2 or MAC address in its new NVE, destination NVE. This message's destination address is the broadcast address. Source NVE receives this message. source NVE should update VM's ARP entry in the central directory at the NVA. Source NVE asks NVA to update its mappings to record IPv4 address of the moving VM along with MAC address of VM, and NVE IPv4 address. An NVE-to-NVA protocol is used for this purpose [RFC8014].

Reverse ARP (RARP) which enables the host to discover its IPv4 address when it boots from a local server [RFC0903] is not used by VMs because the VM already knows its IPv4 address. IPv4/v6 address is assigned to a newly created VM, possibly using Dynamic Host Configuration Protocol (DHCP). Next, we describe a case where RARP is used.

There are some vendor deployments (diskless systems or systems without configuration files) wherein VM users, i.e. end-user clients ask for the same MAC address upon migration. This can be achieved by the clients sending RARP request reverse message which carries the old MAC address looking for an IP address allocation. The server, in this case the new NVE needs to communicate with NVA, just like in the

gratuitous ARP case to ensure that the same IPv4 address is assigned to the VM. NVA uses the MAC address as the key in the search of ARP cache to find the IP address and informs this to the new NVE which in turn sends RARP reply reverse message. This completes IP address assignment to the migrating VM.

All NVEs communicating with this virtual machine uses the old ARP entry. If any VM in those NVEs need to talk to the new VM in the destination NVE, it uses the old ARP entry. Thus the packets are delivered to the source NVE. The source NVE MUST tunnel these in-flight packets to the destination NVE.

When an ARP entry in those VMs times out, their corresponding NVEs should access the NVA for an update.

IPv6 operation is slightly different:

In IPv6, the virtual machine immediately after the move sends an unsolicited neighbor advertisement message containing its IPv6 address and Layer-2 MAC address in its new NVE, the destination NVE. This message is sent to the IPv6 Solicited Node Multicast Address corresponding to the target address which is VM's IPv6 address. NVE receives this message. NVE should update VM's neighbor cache entry in the central directory of the NVA. IPv6 address of VM, MAC address of VM and NVE IPv6 address are recorded in the entry. An NVE-to-NVA protocol is used for this purpose [RFC8014].

All NVEs communicating with this virtual machine uses the old neighbor cache entry. If any VM in those NVEs need to talk to the new VM in the destination NVE, it uses the old neighbor cache entry. Thus the packets are delivered to the source NVE. The source NVE MUST tunnel these in-flight packets to the destination NVE.

When a neighbor cache entry in those VMs times out, their corresponding NVEs should access the NVA for an update.

4.2. Task Migration

Virtualization in L2 based data center networks becomes quickly prohibitive because ARP/neighbor caches don't scale. Scaling can be accomplished seamlessly in L3 data center networks by just giving each virtual network an IP subnet and a default route that points to NVE. This means no explosion of ARP/ neighbor cache in VMs and NVEs (just one ARP/ neighbor cache entry for default route) and there is no need to have Ethernet header in encapsulation [RFC7348] which saves at least 16 bytes.

In L3 based data center networks, since IP address of the task has to change after move, an IP based task migration protocol is needed. The protocol mostly used is the identifier locator addressing or ILA [I-D.herbert-nvo3-ila]. Address and connection migration introduce complications in task migration protocol as we discuss below. Especially informing the communicating hosts of the migration becomes a major issue. Also, in L3 based networks, because broadcasting is not available, multicast of neighbor solicitations in IPv6 would need to be emulated.

Task migration involves the following steps:

Stop running the task.

Package the runtime state of the job.

Send the runtime state of the task to the destination NVE where the task is to run.

Instantiate the task's state on the new machine.

Start the tasks for the task continuing from the point at which it was stopped.

Address migration and connection migration in moving tasks are addressed next.

4.2.1. Address and Connection Migration in Task Migration

Address migration is achieved as follows:

Configure IPv4/v6 address on the target host.

Suspend use of the address on the old host. This includes handling established connections. A state may be established to drop packets or send ICMPv4 or ICMPv6 destination unreachable message when packets to the migrated address are received.

Push the new mapping to hosts. Communicating hosts will learn of the new mapping via a control plane either by participation in a protocol for mapping propagation or by getting the new mapping from a central database such as Domain Name System (DNS).

Connection migration involves reestablishing existing TCP connections of the task in the new place.

The simplest course of action is to drop TCP connections across a migration. Since migrations should be relatively rare events, it is

conceivable that TCP connections could be automatically closed in the network stack during a migration event. If the applications running are known to handle this gracefully (i.e. reopen dropped connections) then this may be viable.

More involved approach to connection migration entails pausing the connection, packaging connection state and sending to target, instantiating connection state in the peer stack, and restarting the connection. From the time the connection is paused to the time it is running again in the new stack, packets received for the connection should be silently dropped. For some period of time, the old stack will need to keep a record of the migrated connection. If it receives a packet, it should either silently drop the packet or forward it to the new location, similarly as in Section 5.

5. Handling Packets in Flight

Source hypervisor may receive packets from the virtual machine's ongoing communications and these packets should not be lost and they should be sent to the destination hypervisor to be delivered to the virtual machine. The steps involved in handling packets in flight are as follows:

Preparation Step It takes some time, possibly a few seconds for a VM to move from its source hypervisor to a new destination one. During this period, a tunnel needs to be established so that the source NVE forwards packets to the destination NVE.

Tunnel Establishment - IPv6 Inflight packets are tunneled to the destination NVE using the encapsulation protocol such as VXLAN in IPv6. Source NVE gets destination NVE address from NVA in the request to move the virtual machine.

Tunnel Establishment - IPv4 Inflight packets are tunneled to the destination NVE using the encapsulation protocol such as VXLAN in IPv4. Source NVE gets destination NVE address from NVA when NVA requests NVE to move the virtual machine.

Tunneling Packets - IPv6 IPv6 packets are received for the migrating virtual machine encapsulated in an IPv6 header at the source NVE. Destination NVE decapsulates the packet and sends IPv6 packet to the migrating VM.

Tunneling Packets - IPv4 IPv4 packets are received for the migrating virtual machine encapsulated in an IPv4 header at the source NVE. Destination NVE decapsulates the packet and sends IPv4 packet to the migrating VM.

Stop Tunneling Packets When source NVE stops receiving packets destined to the virtual machine that has just moved to the destination NVE.

6. Moving Local State of VM

After VM mobility related signaling (VM Mobility Registration Request/Reply), the virtual machine state needs to be transferred to the destination Hypervisor. The state includes its memory and file system. Source NVE opens a TCP connection with destination NVE over which VM's memory state is transferred.

File system or local storage is more complicated to transfer. The transfer should ensure consistency, i.e. the VM at the destination should find the same file system it had at the source. Precopying is a commonly used technique for transferring the file system. First the whole disk image is transferred while VM continues to run. After the VM is moved any changes in the file system are packaged together and sent to the destination Hypervisor which reflects these changes to the file system locally at the destination.

7. Handling of Hot, Warm and Cold Virtual Machine Mobility

Cold Virtual Machine mobility is facilitated by the VM initially sending an ARP or Neighbor Discovery message at the destination NVE but the source NVE not receiving any packets inflight. Cold VM mobility also allows all previous source NVEs and all communicating NVEs to time out ARP/neighbor cache entries of the VM and then get NVA to push to NVEs or get NVEs to pull the updated ARP/neighbor cache entry from NVA.

The VMs that are used for cold standby receive scheduled backup information but less frequently than that would be for warm standby option. Therefore, the cold mobility option can be used for non-critical applications and services.

In cases of warm standby option, the backup VMs receive backup information at regular intervals. The duration of the interval determines the warmth of the standby option. The larger the duration, the less warm (and hence cold) the standby option becomes.

In case of hot standby option, the VMs in both primary and secondary domains have identical information and can provide services simultaneously as in load-share mode of operation. If the VMs in the primary domain fails, there is no need to actively move the VMs to the secondary domain because the VMs in the secondary domain already contain identical information. The hot standby option is the most costly mechanism for providing redundancy, and hence this option is

utilized only for mission-critical applications and services. In hot standby option, regarding TCP connections, one option is to start with and maintain TCP connections to two different VMs at the same time. The least loaded VM responds first and pickup providing service while the sender (origin) still continues to receive Ack from the heavily loaded (secondary) VM and chooses not use the service of the secondary responding VM. If the situation (loading condition of the primary responding VM) changes the secondary responding VM may start providing service to the sender (origin).

8. Virtual Machine Operation

Virtual machines are not involved in any mobility signalling. Once VM moves to the destination NVE, VM IP address does not change and VM should be able to continue to receive packets to its address(es). This happens in hot VM mobility scenarios.

Virtual machine sends a gratuitous Address Resolution Protocol or unsolicited Neighbor Advertisement message upstream after each move.

8.1. Virtual Machine Lifecycle Management

Managing the lifecycle of VM includes creating a VM with all of the required resources, and managing them seamlessly as the VM migrates from one service to another during its lifetime. The on-boarding process includes the following steps:

1. Sending an allowed (authorized/authenticated) request to Network Virtualization Authority (NVA) in an acceptable format with mandatory/optional virtualized resources {cpu, memory, storage, process/thread support, etc.} and interface information

2. Receiving an acknowledgement from the NVA regarding availability and usability of virtualized resources and interface package

3. Sending a confirmation message to the NVA with request for approval to adapt/adjust/modify the virtualized resources and interface package for utilization in a service.

9. Security Considerations

Security threats for the data and control plane are discussed in [RFC8014]. There are several issues in a multi-tenant environment that create problems. In L2 based data center networks, lack of security in VXLAN, corruption of VNI can lead to delivery to wrong tenant. Also, ARP in IPv4 and ND in IPv6 are not secure especially if we accept gratuitous versions. When these are done over a UDP

encapsulation, like VXLAN, the problem is worse since it is trivial for a non trusted application to spoof UDP packets.

In L3 based data center networks, the problem of address spoofing may arise. As a result the destinations may contain untrusted hosts. This usually happens in cases like the virtual machines running third part applications. This requires the usage of stronger security mechanisms.

10. IANA Considerations

This document makes no request to IANA.

11. Acknowledgements

The authors are grateful to Dave R. Worley, Qiang Zu, Andrew Malis for helpful comments.

12. Change Log

- submitted version -00 as a working group draft after adoption
- submitted version -01 with these changes: references are updated, added packets in flight definition to Section 2
- submitted version -02 with updated address.
- submitted version -03 to fix the nits.
- submitted version -04 in reference to the WG Last call comments.

13. References

13.1. Normative References

[RFC0826] Plummer, D., "An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware", STD 37, RFC 826, DOI 10.17487/RFC0826, November 1982, <https://www.rfc-editor.org/info/rfc826>.

[RFC0903] Finlayson, R., Mann, T., Mogul, J., and M. Theimer, "A Reverse Address Resolution Protocol", STD 38, RFC 903, DOI 10.17487/RFC0903, June 1984, <https://www.rfc-editor.org/info/rfc903>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, DOI 10.17487/RFC2629, June 1999, <https://www.rfc-editor.org/info/rfc2629>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <https://www.rfc-editor.org/info/rfc7348>.

[RFC7364] Narten, T., Ed., Gray, E., Ed., Black, D., Fang, L., Kreeger, L., and M. Napierala, "Problem Statement: Overlays for Network Virtualization", RFC 7364, DOI 10.17487/RFC7364, October 2014, <https://www.rfc-editor.org/info/rfc7364>.

[RFC8014] Black, D., Hudson, J., Kreeger, L., Lasserre, M., and T. Narten, "An Architecture for Data-Center Network Virtualization over Layer 3 (NVO3)", RFC 8014, DOI 10.17487/RFC8014, December 2016, <https://www.rfc-editor.org/info/rfc8014>.

13.2. Informative references

[I-D.herbert-nvo3-ila]
Herbert, T. and P. Lapukhov, "Identifier-locator addressing for IPv6", draft-herbert-nvo3-ila-04 (work in progress), March 2017.

Authors' Addresses

Behcet Sarikaya
Denpel Informatique

Email: sarikaya@ieee.org

Linda Dunbar
Huawei USA
5340 Legacy Dr. Building 3
Plano, TX 75024

Email: linda.dunbar@huawei.com

Bhumip Khasnabish
ZTE (TX) Inc.
55 Madison Avenue, Suite 160
Morristown, NJ 07960

Email: vumip1@gmail.com, bhumip.khasnabish@ztetx.com

Tom Herbert
Quantonium

Email: tom@herbertland.com

Saumya Dikshit
Cisco Systems
Cessna Business Park
Bangalore, Karnataka, India 560 087

Email: sadikshi@cisco.com