Network Working Group
Internet Draft
Intended status: Informational
Expires: May 18, 2020

L. Dunbar
Futurewei
B. Sarikaya
Denpel Informatique
B.Khasnabish
Independent
T. Herbert
Intel
S. Dikshit
Aruba-HPE

November 18, 2019

Virtual Machine Mobility Solutions for L2 and L3 Overlay Networks
draft-ietf-nvo3-vmm-06

Abstract

This document discusses Virtual Machine (VM) mobility solutions that are commonly used in overlay-based Data Center (DC) networks. The objective is to describe the solutions and their impact on moving VMs (and applications) from one rack to another connected by the Overlay networks.

For layer 2 networks, it is based on using an NVA (Network Virtualization Authority) - NVE (Network Virtualization Edge) protocol to update the ARP (Address Resolution Protocol) table or neighbor cache entries after a VM (virtual machine) moves from an Old NVE to a New NVE. For Layer 3, it is based on migration of address and connection after the move.

Table of Contents

1. Introduction

This document describes the overlay-based DC networking solutions in support of multi-tenancy and VM mobility. Many large DCs, especially Cloud DCs, host tasks (or workloads) for multiple tenants. A tenant can be a department of one organization or an organization. There is communication among tasks belonging to one tenant and communication among tasks belonging to different tenants or with external entities.

Server Virtualization, which is being used in almost all of today's DCs, enables many VMs to run on a single physical computer or server sharing the processor/memory/storage. Network connectivity among VMs is provided by the network virtualization edge (NVE) [RFC8014]. It is highly desirable [RFC7364] to allow VMs to move dynamically (live, hot, or cold move) from one server to another for dynamic load balancing or optimized workload distribution.

There are many challenges and requirements related to VM mobility in large data centers, including dynamically attaching/detaching VMs to/from Virtual Network Edges (VNEs). In addition, retaining the IP addresses after a move is a key requirement [RFC7364]. Such a requirement is needed in order to maintain existing transport connections.

In traditional Layer-3 based networks, retaining IP addresses after a move is generally not recommended because the frequent move will cause fragmented IP addresses, which complicates IP address management.

In view of many VM mobility schemes that exist today, there is a need to document comprehensive VM mobility solutions that cover both IPv4 and IPv6. Large DC networks can be organized as one large (a) Layer-2 network geographically distributed across buildings/cities or (b) Layer-3 networks with large number of host routes that cannot be aggregated as a result of frequent moves from one location to another without changing the IP addresses.

The connectivity between Layer 2 boundaries can be achieved by the NVE functioning as Layer-3 gateway, performing routing across bridging domain such as in Warehouse Scale Computers (WSC).

2. Conventions used in this document

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:

VM: Virtual Machine

Tasks: Task is a program instantiated or running on a virtual machine or container. Tasks in virtual machines or containers can be migrated from one server to another. We use task, workload 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.

Old NVE: This refers to the old NVE where packets were forwarded to before migration.

New NVE: This refers to the new NVE after migration.

Packets in flight: This refers to the packets received by the Old 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 the systems that are not using configuration files are called end user clients.

Cloud DC: Third party DCs that host applications, tasks or workloads and owned by different organizations or tenants.

3. Requirements

This section states VM mobility requirements on DC networks.

DC networks should support both IPv4 and IPv6 VM mobility.

VM mobility should not require changing their IP addresses after the move.

There exist "Hot Migration" where transport service continuity is maintained, and "Cold Migration" where the transport service needs to be restarted, i.e., execution of the tasks is stopped on the "Old" NVE, moved to the "New" NVE and the task is restarted.

VM mobility solutions/procedures should minimize triangular routing except for handling packets in flight.

VM mobility solutions/procedures should not need to use tunneling except for handling packets in flight.

4. Overview of the VM Mobility Solutions

Layer-2 and Layer-3 mobility solutions are described respectively in the following sections.

4.1. VM Migration in Layer-2 Network

Ability to move VMs dynamically, from one server to another, makes it possible for dynamic load balancing or workload distribution.

Therefore, this scheme is highly desirable for utilization in large scale multi-tenant DCs.

In a Layer-2 based VM migration approach, a VM that is moving to another server does not change its IP address. But since this VM is now under a new NVE, previously communicating NVEs will continue sending their packets to the Old NVE. To solve this problem, Address Resolution Protocol (ARP) cache in IPv4 [RFC0826] or neighbor cache in IPv6 [RFC4861] in the NVEs need to be updated promptly. All NVEs need to change their caches associating the VM Layer-2 or Medium Access Control (MAC) address with the new NVE's IP address as soon as the VM moves. Such a change enables all NVEs to encapsulate the outgoing MAC frames with the current target NVE IP address. It may take some time to refresh the ARP/ND cache when a VM has moved to a New NVE. During this period, a tunnel is needed for that Old NVE to forward packets destined to the VM under the New NVE.

In case of IPv4, immediately after the move, the VM should send a gratuitous ARP request message containing its IPv4 and Layer-2 MAC address to its new NVE. This message's destination address is the broadcast address. Upon receiving this message, both old and new NVEs should update the VM's ARP entry in the central directory at the NVA, to update its mappings to record the IPv4 address and MAC address of the moving VM along with the new 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. Next, we describe a case where RARP is used.

There are some vendor deployments (e.g., diskless systems or systems without configuration files) where the VM's user, i.e., end-user client asks for the same MAC address upon migration. This can be achieved by the clients sending RARP request message which carries the 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 message. This completes IP address assignment to the migrating VM.

Other NVEs communicating with this VM could have the old ARP entry. If any VMs in those NVEs need to communicate with the VM

attached to the new NVE, old ARP entries might be used. Thus, the packets are delivered to the old NVE. The old NVE MUST tunnel these in-flight packets to the new NVE.

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

IPv6 operation is slightly different:

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

Other NVEs communicating with this VM might still use the old neighbor cache entry. If any VM in those NVEs need to communicate with the VM attached to the new NVE, it could use the old neighbor cache entry. Thus, the packets are delivered to the old NVE. The old NVE MUST tunnel these in-flight packets to the new 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 in Layer-3 Network

Layer-2 based DC networks become quickly prohibitive because ARP/neighbor caches don't scale. Scaling can be accomplished seamlessly in Layer-3 data center networks by just giving each virtual network an IP subnet and a default route that points to its NVE. This means no explosion of ARP/ neighbor cache in VMs and NVEs (just one ARP/ neighbor cache entry for the default route) and there is no need to have Ethernet header in encapsulation [RFC7348] which saves at least 16 bytes.

Even though the term VM and Task are used interchangeably in this document, the term Task is used in the context of Layer-3 migration mainly to have slight emphasis on the task of moving an entity that is instantiated on a VM or a container.

Traditional Layer-3 based DC networks require IP address of the task to change after moving because the pre-fixes of the IP address usually reflect the locations. It is necessary to have an IP based VM migration solution that can allow IP addresses staying the same after the VMs move to different locations. The Identifier Locator Addressing or ILA [I-D.herbert-nvo3-ila] is one of such solutions.

Because broadcasting is not available in Layer-3 based networks, multicast of neighbor solicitations in IPv6 would need to be emulated.

Cold task migration, which is a common practice in many data centers, involves the following steps:

- Stop running the task.
- Package the runtime state of the job.
- Send the runtime state of the task to the new NVE where the task is to run.
- Instantiate the task's state on the new machine.
- Start the tasks continuing it from the point at which it was stopped.

Address migration and connection migration in moving tasks or VMs 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 Task.
- Suspend use of the address on the old Task. 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 VM. Communicating VMs will learn of the new mapping via a control plane either by participating 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 all TCP connections to the VM across a migration. If the migrations are relatively rare events in a data center, impact is relatively small when TCP connections are 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 approach 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 could 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 can either silently drop the packet or forward it to the new location, as described in Section 5.

5. Handling Packets in Flight

The Old NVE may receive packets from the VM's ongoing communications. These packets should not be lost; they should be sent to the New NVE to be delivered to the VM. 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 Old NVE to a New NVE. During this period, a tunnel needs to be established so that the Old NVE can forward packets to the New NVE. Old NVE gets New NVE address from NVA in the request to move the VM. The Old NVE can store the New NVE address for the VM with a timer. When the timer expired, the entry for the New NVE for the VM can be deleted.

Tunnel Establishment - IPv6: Inflight packets are tunneled to the New NVE using the encapsulation protocol such as VXLAN in IPv6.

Tunnel Establishment - IPv4: Inflight packets are tunneled to the New NVE using the encapsulation protocol such as VXLAN in IPv4.

Tunneling Packets - IPv6: IPv6 packets received for the migrating VM are encapsulated in an IPv6 header at the Old NVE. New NVE decapsulates the packet and sends IPv6 packet to the migrating VM.

Tunneling Packets - IPv4: IPv4 packets received for the migrating VM are encapsulated in an IPv4 header at the Old NVE. New NVE decapsulates the packet and sends IPv4 packet to the migrating VM.

Stop Tunneling Packets: When the Timer for storing the New NVE address for the VM expires. The Timer should be long enough for all other NVEs that need to communicate with the VM to get their NVE-VM cache entries updated.

6. Moving Local State of VM

In addition to the VM mobility related signaling (VM Mobility Registration Request/Reply), the VM state needs to be transferred to the New NVE. The state includes its memory and file system if the VM cannot access the memory and the file system after moving to the New NVE. Old NVE opens a TCP connection with New 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 New NVE should find the same file system it had at the Old NVE. Pre-copying 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 New NVE Hypervisor which reflects these changes to the file system locally at the destination.

7. Handling of Hot, Warm and Cold VM Mobility

Both Cold and Warm VM mobility (migration), refers to the VM being completely shut down at the old NVE before restarted at the new NVE. Therefore, all transport services to the VM need to restart.

Upon starting at the new NVE, the VM should send an ARP or Neighbor Discovery message. Cold VM mobility also allows the Old NVE and all communicating NVEs to time out ARP/neighbor cache entries of the VM. It is necessary for the NVA to push the updated ARP/neighbor cache entry to NVEs or for NVEs to pull the updated ARP/neighbor cache entry from NVA.

The Cold VM mobility can be facilitated by cold standby entity receiving scheduled backup information. The cold standby entity can be a VM or other form factors which is beyond the scope of this document. The cold mobility option can be used for non-critical applications and services that can tolerate interrupted TCP connections.

The Warm VM mobility refers the backup entities receive backup information at more frequent intervals. The duration of the interval determines the warmth of the option. The larger the duration, the less warm (and hence cold) the Warm VM mobility option becomes.

There is also a Hot Standby option in addition to the Hot Mobility, where there are VMs in both primary and secondary NVEs. They have identical information and can provide services simultaneously as in load-share mode of operation. If the VM in the primary NVE fails, there is no need to actively move the VM to the secondary NVE because the VM in the secondary NVE already contains identical information. The Hot Standby option is the costliest mechanism, 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 starts providing service while the sender (origin) still continues to receive Ack from the heavily loaded (secondary) VM and chooses not to use the service of the secondary responding VM. If the situation (loading condition of the primary responding VM) changes the secondary VM may start providing service to the sender (origin).

8. VM Operation

Once a VM moves to a new NVE, the VM's IP address does not change and the VM should be able to continue to receive packets to its address(es).

The VM needs to send a gratuitous Address Resolution message or unsolicited Neighbor Advertisement message upstream after each move.

The VM lifecycle management is a complicated task, which is beyond the scope of this document. Not only it involves monitoring server utilization, balancing the distribution of workload, etc., but also needs seamless management VM migration from one server to another.

9. Security Considerations

Security threats for the data and control plane for overlay networks are discussed in [RFC8014]. There are several issues in a multi-tenant environment that create problems. In Layer-2 based overlay DC networks, lack of security in VXLAN, and corruption of VNI can lead to delivery of information to the wrong tenant.

Also, ARP in IPv4 and ND in IPv6 are not secure, especially if we accept the gratuitous versions. When these are done over a UDP encapsulation, as in VXLAN, the problem gets worse since it is trivial for a non-trusted entity to spoof UDP packets.

In Layer-3 based overlay data center networks, the problem of address spoofing may arise. An NVE may have untrusted tasks attached to it. This usually happens in situations when the VMs (tasks) running third party applications. This requires the usage of stronger security mechanisms.

10. IANA Considerations

This document makes no request to IANA.

11. Acknowledgments

The authors are grateful to Bob Briscoe, David Black, Dave R. Worley, Qiang Zu, and 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,
  o 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.

. Submitted version - 05 to address IETF LC comments from TSV area.

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, March 1997.

[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

Linda Dunbar
Futurewei
Email: ldunbar@futurewei.com

Behcet Sarikaya
Denpel Informatique
Email: sarikaya@ieee.org

Bhumip Khasnabish
Independent
Email: vumip1@gmail.com

Tom Herbert
Intel
Email: tom@herbertland.com

Saumya Dikshit
Aruba-HPE
Bangalore, India
Email: saumya.dikshit@hpe.com