International Journal of Scientific Research in Computer Science and Engineering
Vol.6, Issue.2, pp.64-71, April (2018)

Research Paper

E-ISSN: 2320-7639

An Approach to Network Level Support for Inter-Cloud Live Migration

Supreet Kaur

Department of Computer Engineering and Technology, Guru Nanak Dev University, Amritsar, Punjab, India

*Corresponding Author: supreetk033@gmail.com

Available online at: www.isroset.org

Received 12/ Mar/2018, Revised 18/Mar/2018, Accepted 02/Apr/2018, Online 30/Apr/2018

***Abstract*—** Inter-Cloud live migration of virtual machines is beneficial to a number of scenarios to keep communications between and to virtual machines uninterrupted, inter-cloud live migration must rely on support from the network layer to handle the dynamic relocation of virtual machines between different networks. We developed a prototype implementation of these mechanisms for the ViNe virtual network developed at University of Florida. ViNe provided the methods that simplify resource management and deal with connectivity constraints and support legacy applications for distributed systems by enabling global address ability of VN-connected machines through either a common layer 2 Ethernet or a NAT-free layered 3 IP networks.

***Keywords*—** ViNe, IPOP, TAP device, ViNe Overlay, HWDT

1. INTRODUCTION TO INTER-CLOUD LIVE MIGRATION

A mechanism for inter-cloud live migration of virtualization systems is disclosed, method of the invention includes receiving notification that live migration of a virtual machine (VM) has completed.

In the Virtual Machine (VM) is migrated from a source host computing machine on a source cloud of a target host computing machine on a target cloud, receiving requests sent to a previous IP address of the VM associated with the source cloud, the requests routed over a layer 2 network tunnel established between the source cloud and the target cloud.

Configuration new network interface with a new IP address for the VM to receive requests directly via a communication connection between the target cloud and simultaneously handling the requests for both of the previous IP addresses received via the layer 2 network tunnel.

The new IP address via the communication connection between the target cloud.

1.1. OBJECTIVES OF INTER-CLOUD LIVE MIGRATION

The goal is to relocate a virtual machine up and running in one cloud (data center) to another without losing VM's state.

1.2. MOTIVATION OF INTER-CLOUD LIVE MIGRATION

a) It allows a clean separation between hardware and software.

b) Facilitates fault management, load balancing and low level system maintenance.

c) Data center can redirect traffic at peak time.

1.3. FEATURES OF INTER-CLOUD LIVE MIGRATION

a) It allows a clean separation between hardware and software.

b) Facilitates fault management, load balancing and low level system maintenance.

c) Data center can redirect traffic at peak time.

1.4. APPLICATIONS OF INTER-CLOUD LIVE MIGRATION

a) Cloud Bursting:- Cloud Bursting is an application deployment model in which an application runs in a private cloud or data center and bursts into a public cloud when the demand for computing capacity spikes and the advantage of hybrid cloud deployment is that an organization only pays for extra compute resources when they are needed.

b) Enterprise IT consolidation:- Convert Physical System to Virtual Machines, Leverage Live Migration and Enable High Availability (HA), Continuous Availability, Automated Resource Management.

1.5. ARCHITECTURE OF INTER-CLOUD LIVE MIGRATION

Suppose VM1 has to be migrated from clouding A to cloud B

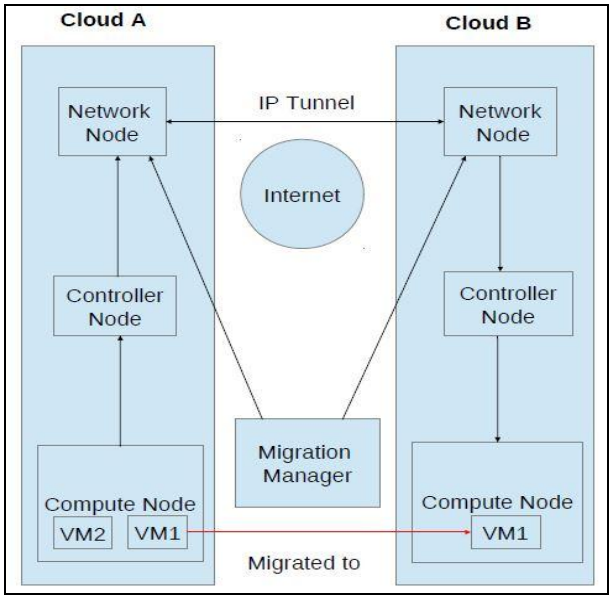

Figure 1. Architecture of Inter-Cloud Live Migration

The moment a migration processed is triggered:-

a) The Cloud A's controller will create a snapshot of VM1.

b) This memory picture is transferred through the network controller node in an IP tunnel over Internet.

c) The Cloud B's controller node spawns a new VM based on the memory snapshot received.

d) We reconfigure the router of Cloud A's network node to redirect VM1's traffic to Cloud B's network controller.

2. RELATED WORK

It is essential that VMs keep their network address (i.e. IP) unchanged after migrating from one host server to another. Important system and applications state information can be lost if the IP address of VMs change, resulting in loss of data and compute cycles. The Internet was not designed to support nodes moving between subnets, but with the popularity of mobile devices and advances in VM migration technologies, mobility protocols and solutions have been proposed and developed. **[1]** The issues created by inter-cloud live migration are equivalent to the problem of device mobility between different IP networks. An overview of existing solutions is presented in the followings:-

a) Mobile IP: Developed by the Internet Engineering Task Force (IETF), Mobile IP is a standard mechanism providing support for roaming of mobile devices on IPV4 and IPV6 networks. Each Mobile device is assigned a permanent IP address known as home address, belonging to its home network. When a mobile device joins a new network (called a foreign network), it receives a new IP address belonging to this network: its care-of address. Care-of addresses are announced by an entity called the foreign agent, residing in the foreign network.

b) Overlay Networks: Overlay networks are peer-to-peer networks built top of a physical network infrastructure (e.g. The Internet). The overlay networks route communication between the nodes of the virtual IP networks. Overlay networks of mobility support to require an overlay network stack to be installed and running on every node of a virtual cluster. The overlay network traffic is fully controlled within VMs with virtual IP assignments that are independent from the network interface accessing the Internet. Decoupling the low-level IP network of the higher level IP networks to allow support mobility.

c) Virtual Private Networks (VPN): A VPN is a network that is constructed using public wires usually the Internet to connect to a private network, such as company's internal network. There are a number of systems that enable you to create networks using the Internet as the medium for transporting data. Another type of migration supported to rely on the presence of VPN connecting multiple sites. These VPNs unify multiple network together. However, VPNs with reconfiguration capabilities are not always available, or they restrict their reconfiguration to network administrators.

d) ARP and Tunneling: The Address Resolution Protocol (ARP) is used to discover mappings between network layer addresses (IP addresses) and link layer addresses (MAC addresses for Ethernet). Proxy ARP is a mechanism in which a host answers ARP requests on behalf of a host not present at the network. This

technique is increasingly used in inter-cloud live migration, in order to capture traffic intended for a migrated virtual machine. An advantage of this mechanism is that it is supported by most operating systems that could be deployed in a cloud. A clear advantage of using ViNe is that Internet connectivity recovery problems have already been addressed.

e) **Hypervisor-Based:** Typically, all network traffic to and from a VM goes through the hosting hypervisor. Thus, the hypervisor is a good place to implement the necessary network support for VM migration. The drawbacks of this approach from a multi-cloud scenario are the dependency on the hypervisor (Implementing the migration support) and the need for an all-to-all communication between hypervisors (not necessarily available).

3. METHODOLOGY

3.1. INTER-CLOUD NETWORK CONNECTION

ViNe is user-level virtual network software developed at the University of Florida. The ViNe network consists of a set of Virtual Networks (VNs). Each VN consists of a set of Virtual Routers (VRs). Any machine deployed with ViNe software can serve as a VR and VRs in the same VN have network connection with each other. **[2]** We transform the elements of 'VirtualRouter' and the elements of 'Server' to the elements of 'VM'. Every time an element of 'VM' is created on the customized model, an element of 'VirtualRouter' will be created on the customized model, an element of 'VirtualRouter' will be created on the runtime model of ViNe and the ViNe software will be configured automatically on this VM. **[3]** ViNe is being implemented as user-level software since it is easier to develop, test and deploy compared to Kernel-based systems.

3.2. ARCHITECTURE OF VINE

ViNe is a virtual network system that aims to provide all-to-all connectivity in complex network infrastructures, such as those found in grids and clouds. These infrastructures can include networks using private IP addressing or firewalls to control access to resources. ViNe can support multiple isolated virtual networks sharing the same overlay. ViNe overlays have been used to successfully connect VMs on different cloud providers to run a large scale bioinformatics application.

The architecture of ViNe is based on a user-level software router. Software router processes are in charge of creating and maintaining the network overlay, and tunneling traffic between physical networks. We refer to a node running the ViNe software as a ViNe router (VR). Each VR act as a gateway to machine connected to the same sub-network.

ViNe is flexible in terms of configuration, which can be dynamically changed by modifying the VRs operating parameters.

Routing decisions on the ViNe overlays are based on tables managed by each VR. Two types of tables are maintained by VRs:

a) Local Network Description Tables (LNDTs)
b) Global Network Description Tables (GNDTs)

Each LNDT lists the nodes a VR is responsible for nodes on the same sub-network that are allowed to participate in the ViNe virtual network.

GNDTs store information about the physical networks and VRs participating in a ViNe virtual network like a network routing table, it contains necessary data to forward network packets of their correct destinations.

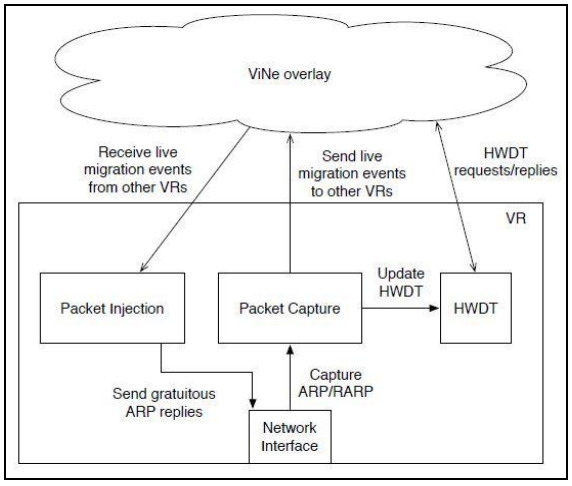

Figure 2. Architecture of ViNe

In Figure 2 show diagram, the overall architecture of the implementation of our mechanisms in ViNe. In addition to the GNDT and LNDT, a new table is managed by each ViNe router: Hardware Description Table (HWDT). To populate and maintain this table, the packet capture module is extended to capture all ARP and RARP traffic arriving on the physical interface. This allows each ViNe router to be aware of the MAC addresses and IP addresses present on the network.

When a migration is detected (for instance when capturing a RARP packet), the VR first queries other VRs to discover the IP address of the migrated virtual machine. These queries are replied by the HWDT of each VR. Then,

the migration event is propagated to other VRs in the virtual network.

When a live migration event is received from another VR, the packet injection module builds gratuitous ARP replies and injects them on the physical network. Additionally, the GNDT and LNDT of each VR is updated.

ViNe can be deployed in different scenarios described in the following subsections. For each scenario, networking mechanisms to support live migration is discussed:

I. Environments with heavy network protection:
Network protections filter layer 2 communication, making ARP related mechanisms unusable. To circumvent this limitation, all VMs are required to become VRs. To support migration, all VMs should be configured with an additional network interface configured with a virtual IP address. ViNe management actions can be triggered by the cloud middleware when a VM migration is initiated.

II. Protection free Clouds:
In such environments ViNe can be deployed in its recommended mode of operation: one VR in each domain acts as a gateway to overlay networks. When a VM migrates from one domain to another, the VR in the origin subnet will use ARP messages to redirect and receive packets destined to the moving VM for appropriate tunneling. This way, VRs can capture packets of overlaying routing, and at the same time get out of the way for local communication even among 'native' and 'foreign' VMs.

III. Public Subnets:
ViNe can be deployed to connect VMs deployed on public and private subnets. Moving VMs with public IP addresses is challenging on about support of the physical infrastructure to change the physical routes. However, a migrated VM will always depend on VR routing to communicate with 'native' nodes on a 'foreign' subnet.

3.3. ViNe Deployment

The ViNe overlayed is used to tunnel packets, over UDP or TCP connections.

In Figure 3 show diagrams, illustrates a ViNe deployment. Two physical networks are 1.2.3.0/24 and 192.168.0.0/24, are participating in a ViNe virtual network. Each network contains a VR acting as a gateway to the local network and the rest of the virtual network. Each VR has a copy of the GNDT and of its corresponding LNDT (LNDT A for VR A, LNDT B for VR B).

When a packet is sent from 1.2.3.1 to 192.168.0.1, traffic is routed through VR A. First, VR A verifies that the source host is authorized to participate in the VN, by checking its LNDT. Since 1.2.3.1 is present in the LNDT, the packet is authorized.

Then, VR A consults its GNDT like a routing table, the most specific network entry is selected (in this case 192.168.0.0/24) and the packet is forwarded to the corresponding VR, using the public address of VR B, 4.5.6.7. Once received by VR B, the packet is delivered to 192.168.0.1.

Even though network B is using NAT with private IP addresses, routing packets of the level of the virtual network allows all-to-all connectivity between nodes from networks A and B.

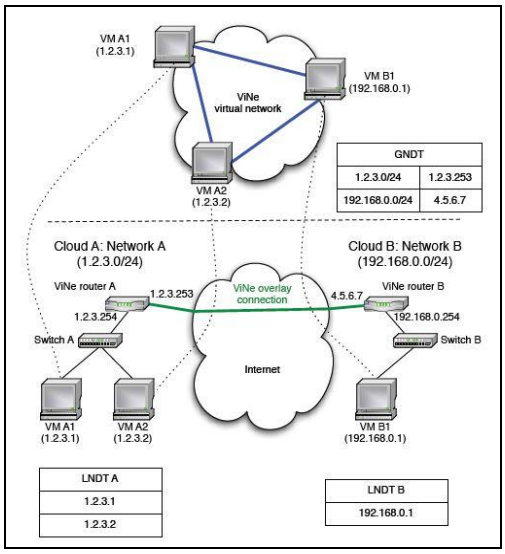

Figure 3. Example deployment of ViNe

The configuration information (GNDT and LNDT) can be updated dynamically by exchanging information about VRs. This allows ViNe to react to changes in the network environment. This is the case of virtual machine live migration.

Additionally, the LNDTs of the source and destination VRs need to be updated. The migrated IP is removed from the source VR LNDT and added to the destination VR LNDT.

Figure 4 illustrates the state of the ViNe deployment presented in Figure 3. After live migration the virtual machine with IP address 1.2.3.2 from clouding A to cloud B, the GNDT is modified to include a host-only network associated with VR B. Additionally; 1.2.3.2 is removed from LNDT A and added to LNDT B.

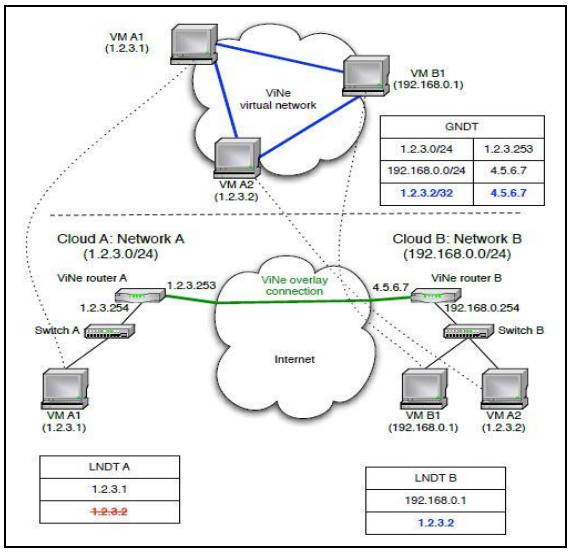

Figure 4. ViNe deployment after live migration

3.4. VIRTUAL NETWORK MODELS

Traditionally, VNs provide either an interface model, where there exists one VN software stack per host, or the router model, where there exists one VN software stack per some subset of machines in an LAN. **[4]** We also present a new model, hybrid that bridges the gap between the two models by taking a subset of features from both.

In this section, we focus on the basic architecture of the VNs leaving the system independent networking features on the following:-

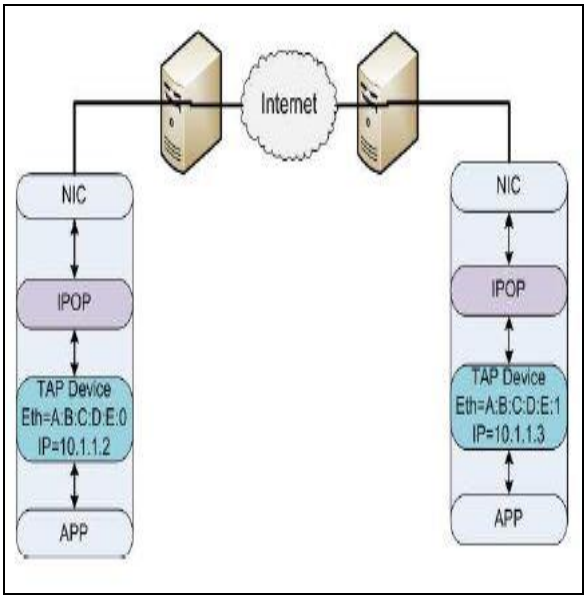

Figure 5. A VN deployed as an interface for single machine usage. The user of the machine is presented two interfaces on two different IP subnets. All non-VN subnet based traffic is routed normally via the default interface.

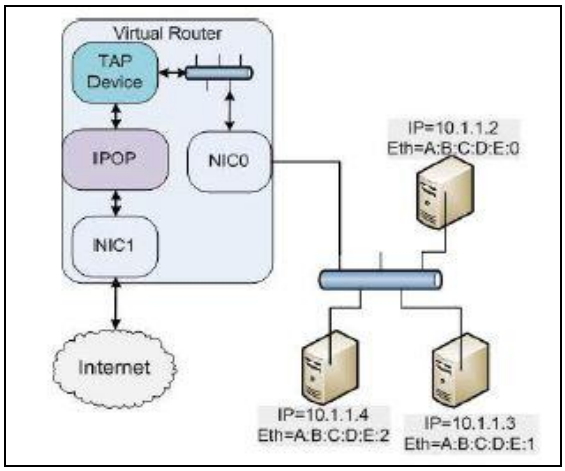

Figure 6. A VN deployed as a router providing virtual network access for an entire layer 2 network. Each machine in the network only has a VN-based address, though they can communicate directly with each other. The machine hosting the VN can also have an IP address in the network by assigning one to the bridge.

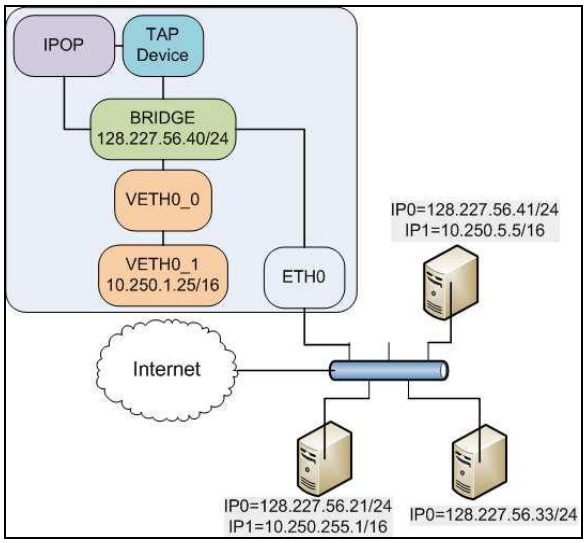

Figure 7. A VN deployed in a hybrid model providing virtual network access for a single machine but bypassing the VN when a VN peer is local. This model is similar to having two network cards from a single machine going to one switch. The key feature is that this model allows a machine to be in multiple IP address space subnets and have layer 2 traffic as well.

The prime factor which the three designs share is the TAP device, a Virtual Ethernet device that is available for almost all modern operating systems, including Windows, Linux, Mac OS/X, BSD and Solaris.

The TAP device functions by letting software write incoming packets of the device which is treated as packets that came from over the network. Packets that are written to the TAP device by OS sockets are available by reading the TAP device. The VN

receives incoming packets of the overlay and writes them to the TAP device and receives outgoing packets of the TAP device and sends them through the overlay.

IPOP (IP-over-P2P) is an open source user-centric software virtual network allows the end users to define and create their own virtual private networks (VPNs). **[5]** However, IPOP uses the P2P address for message routing which is decoupled from the physical network address of a host, virtual IP address migration happens transparently to the application (even across domains), and connectivity can be established as more quickly as it takes for P2P links to be re-established.

Table 1. Qualitative comparison of the three deployment models

Features	Interface	Router	Hybrid
Host LAN	No Assumption	Ideally, VLAN	No Assumption, though may have duplicate address allocation in the same subnet for different namespaces
Host Software	IPOP, TAP	End node: None Router: IPOP, TAP, Bridge	IPOP, TAP, Virtual Ethernet, bridge
Host Overhead	CPU, Memory	End node: none Router: CPU, Memory	CPU, Memory
LAN traffics	Through IPOP	Bypasses IPOP	Bypasses IPOP
Migration	Handled by node	Involves source and target routers	Handled by node
Tolerance to Faults	Nodes is independent	Router fault affects all LAN nodes	Nodes is independent

3.5. LIVE MIGRATION SUPPORT IN VINE

The design and prototype implementation of the mechanisms we proposed to add virtual machine live migration supported to ViNe. These mechanisms rely on ARP and tunneling techniques to allow virtual machines to migrate between two different clouds of interrupting their communications.

Our live migration support in ViNe by describing the following mechanisms:

a) Automatic Detection of Live Migration:- By automatically detecting live migration events, ViNe does not require bidding with the cloud management stacks, making it more independent and portable.

When a live virtual machine migration occurs, the destination hypervisor sends Ethernet frames on behalf of the migrated virtual machine in order to reconfigure the physical infrastructure. Both KVM and VMware ESX send RARP packets when a migrated virtual machine resumes execution, while Xen sends ARP packets. We modified the ViNe routing software to capture these packets in order to discover newly migrated virtual machines. RARP packets provide VRs with the MAC addresses of migrated virtual machines. To reconfigure the VN, additional information is required.

b) Virtual Network Reconfiguration:- After detecting live migration events, the virtual network is reconfigured to learn the position of the migrated VM.

After a live migration has been detected, the VN needs to be reconfigured to route to the correct sub-network any traffic originating sent or received by the migrated VM. First, the IP address of the migrated VM is discovered. When a VR detects a live migration, it contacts other VRs in the overlay network in order to discover the IP address of the migrated VM. Each VR continuously maintains a table mapping IP and MAC addresses of VM connected to its physical sub-network, allowing to reply to such requests. This table is

c) ARP-Based traffics redirection:- After migration, ARP mechanisms are used to redirect traffic that the migrated VM sends and receives.

To be able to capture this traffic, the VRs need to become the link layer destination address of such packets. The link layers destination address of an IP packet is determined through the ARP protocol. This information is stored on each machine in a structure called an ARP cache. To redirect traffic to the VR, ARP caches of the VMs need to be reconfigured. To updates these caches, gratuitous ARP replied packets are used. Gratuitous ARP replies are responses to ARP requests that were never made, broadcasting information telling the machines on the network that an IP address is reachable through a new link layer address. In particular, gratuitous ARP packets are used in high-availability environment to perform failover.

d) Routing Optimization:- The mechanisms allow a VM to be migrated between different networks of

interrupting its communication. However, communication is non optimal. Any traffic between the migrated VM and hosts of the destination network goes through the destination site VR. This is inefficient compared to direct communication. Moreover, migration can be prompted by a need for faster communication between the migrated VM and hosts of the destination network.

Using proxy ARP, VRs respond to such requests for their own MAC addresses, and forward traffic to the correct destination using the ViNe infrastructure. When a VM migration occurs, we use the same aforementioned mechanisms. Additionally, the destination network VR sends gratuitous ARP replies from each VM on its LAN, in order to update their ARP cache. This enables direct communication between link layer between VMs of the destination site, which allows to use the full available network bandwidth.

4. IMPLEMENTATION AND RESULTS OF NETWORK LEVEL SUPPORT FOR LIVE MIGRATION

An Implement of ViNe allowed VMs to freely move independently of the destination subnet in which effectively enabling live migration of VMs over multiple clouds and something difficult to accomplish with the traditional problems.

The self-configuring system must be transparent to the environment and required no interaction from the administrator besides starting the system. Live Migration supported Networking Protocols such as VNs, it means the ability to join a network of explicitly adding network rules or routing tables.

Furthermore, the results of Networking protocols support for Inter-Cloud Live Migration and features are as follows:-

a) Layer 3 Communication in a LAN:- IP is a layer 3 protocol. Layer 2 devices such as switches, bridges and hubs are not aware of IP addresses. When a layer 3 packet is being sent on a layer 2 networks, the ARP is used to determine the layer 2 address. In a typical IP subnet, all machines talk directly with each other through switches. In our Virtual Network Model, we aim to provide a large, flat subnet spanning across all nodes connected to the VN.

b) Allocating Addresses:- IP addresses are traditionally allocated to one of three ways:

i. Statically
ii. Dynamically through DHCP
iii. Through pseudo-random link local addressing.

In our model, we focus on static and dynamic addressing. DHCP as defined in the RFCs enables dynamic configuration of addresses, routing and other networking related features. While many different clients and servers exist, they all tend to support the basic features of allowing the server to specify to a client and IP address, a gateway address, and domain name servers.

c) Domain Name Servers and Services:- Name services allow machines to be addressed with names that are more meaningful to users than numeric addresses. To support DNS, this requires that the OS was programmed with the VN's DNS servers IP, which we take generically to be the lowest available IP address in a subnet. In static configuration, this process requires the user to manually add this address, though through DHCP this is set automatically.

d) Supporting Migration:- Apart from advantages like performance isolation, security, and portability, one of the significant advantages of using VMs is the capability to migrate the VM with its entire software stack from one physical host of another. This migration may be performed in a stop-start manner, where the VM is paused, migrated to another host and restarted, or in a live mode, which attempts to minimize down time to reduce interruption to services running on the VM.

Unlike interface and hybrid models, the VN router does not support one-to-one mapping and VN router tends to have one P2P address for many IP addresses.

The VN interface and hybrid models to support migration of the virtual address; this is a product of the decentralized, isolated to overlay approach where each overlays end point has an one-to-one mapping to VN end point, e.g. P2P to IP.

Two subnets (X and Y) with Virtual Router, Two VMs on subnet X (VMX and VMY) and one VM on subnet Y (VMZ) were initially created; all pairs of VMs establish communication between with each other. VMX and VMY with direct communication and VMY and VMZ with ViNe infrastructure support Inter-Cloud Live Migration. After successfully found migration VMX to subnet Y without any communication interruption, inspection of ARP cache and DNS confirmed that VMX and VMZ established direct communication in subnet Y, while VMX and VMY begin communicating with ViNe infrastructure support Inter-Cloud Live Migration.

5. CONCLUSION AND FUTURE SCOPE

The design of mechanisms to support inter-cloud live migration of virtual machines by reconfiguring virtual network infrastructures by using ARP based techniques and leveraging ViNe tunneling and reconfiguration capabilities, ViNe is able to keep network traffic uninterrupted. Our mechanisms detect live migration automatically from messages broadcast on the network, making them independent of the cloud management infrastructure. It will measure several metrics: inter-cloud and intra-cloud network communication before and after live migration, detection time of live migration and reconfiguration time. By allowing efficient inter-cloud live migration, execution platforms can be relocated between multiple clouds.

REFERENCES

[1] Mauricio Tsugawa, Pierre Riteau, Andrea Matsunaga, Jose Fortes, "*User-level Virtual Networking Mechanisms to Support Virtual Machine Migration Over Multiple Clouds*", IEEE International Workshop on Management of Emerging Networks and Services, pp. 568-572, 2010

[2] Bo An, Zhicheng Cui, Ying Zhang, "*Towards a Model-Defined Cloud-of-Clouds*", Services Transactions of Cloud Computing, Volume. 4, Issue. 2, pp. 1-14, 2016

[3] Pierre Riteau, "*Building Dynamic Computing Infrastructures over Distributed Clouds*", International Parallel Distributed Processing Symposium, pp. 1-5, 2011

[4] David Isaac Wolinsky, Yonggang Liu, "*On the Design of Scalable, Self-Configuring Virtual Networks*", ACM, Volume. 9, No. 11, pp. 1-12, 2009

[5] A. Ganguly, R. Figueiredo, "*IP over P2P: Enabling self-configuring virtual IP networks for grid computing*", International Parallel and Distributed Processing Symposium, Volume. 4, Issue. 2, pp. 1-10, 2016

Authors Profile

"Supreet Kaur" received her under-graduate degree from Amritsar College of Engineering and Technology (ACET), Amritsar (Punjab), India. She is currently candidate of M.Tech. CSE (Computer Science and Engineering) at Department of Computer Engineering and Technology (CET), Guru Nanak Dev University (GNDU), Amritsar (Punjab), India. Her main research work focuses on Cloud Computing, Inter-Cloud, Artificial Intelligence, Nature Inspired Metaheuristic Algorithms.

