EFFICIENT VIRTUAL MACHINE MOBILITY IN CLOUD COMPUTING

By

Guruprasad K Rao
A Thesis
Submitted to the
Graduate Faculty
of
George Mason University
in Partial Fulfillment of
The Requirements for the Degree
of
Master of Science
Computer Engineering

Committee:

________________ Dr. Bijan Jabbari, Thesis Director

Mulpuri, V. Rao ________ Dr. Mulpuri Rao, Committee Member

________________ Dr. Jeremy E. Allnutt, Committee Member

________________ Dr. Andre Manitius, Chairman, Department of Electrical and Computer Engineering

________________ Dr. Lloyd J. Griffiths, Dean, Volgenau School of Engineering

Date: 12/08/2011 ________

Fall Semester 2011
George Mason University
Fairfax, VA

Efficient Virtual Machine Mobility in Cloud Computing

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at George Mason University

By

Guruprasad Rao K
Bachelor of Engineering
Vishweshwariaya Technological University, 2007

Director: Dr. Bijan Jabbari, Professor
Department of Electrical and Computer Engineering

Summer Semester 2011
George Mason University
Fairfax, VA

Dedication

This is dedicated to my parents and my niece.

Acknowledgement

I would like to express my sincere gratitude to my advisor Dr. Bijan Jabbari. Without his patient guidance and support, none of this would have been achieved. I would also like to thank Dr. Mulpuri Rao and Dr. Jeremy E. Allnut for their valuable inputs and suggestions towards my thesis. Last, but not least, I would like to thank my parents for their encouragement and trust.

Table of Contents

List of Figures

Abstract

EFFICIENT VIRTUAL MACHINE MOBILITY IN CLOUD COMPUTING

Guruprasad Rao K

George Mason University, 2011

Thesis Director: Dr. Bijan Jabbari

Cloud computing change the Internet into a new computing platform, is a business model that achieve purchase on-demand and pay-per-use in network, has a broad development prospects. Virtualization technology is a core technology of cloud computing, the virtual machine is the basic unit of the cloud computing platforms, cloud providers providing services to clients by virtual machines must ensure the security and isolation. The capability to migrate applications with no perceivable effect to the end user enables service providers to develop new and improved methods for provisioning and maintaining data center infrastructure. Cloud computing providers can now perform hardware maintenance, consolidate CPU and memory resources, or migrate mission-critical applications from a data center when necessary without affecting the service-level agreements (SLAs) of the applications. A cumbersome issue is dealing with the network deployment context required to preserve end-to-end connectivity as VMs are migrated to new physical machines that are likely located on different networks. To the best of our knowledge, all proposed ideas rely on ‘virtual networks’ (i.e., overlay networks) to facilitate the interconnection of VMs. We take a radically different approach. The significant growth of Internet traffic and increase of routing tables require solutions to address Internet scalability and resiliency. The approach chosen is to move away from the flat legacy Internet routing to hierarchical routing, separating edge networks from transit networks. In the context of Cloud computing and Internet datacenter, Id/locator splitting provides a convenient method for establishing the network context for a VM that is migrated to a new physical machine. A protocol, the Location Identifier Separation Protocol (LISP) follows this paradigm, adding in particular a control-plane based on a mapping system, which separates the location function from the identification function of the IP address.

OpenLISP is an open source implementation of the LISP proposal in the FreeBSD OS. OpenLISP provides complete Data Plane support for both encapsulation and decapsulation operations, for IPv4 and IPv6, as well as EID-to-RLOC mapping storage and efficient lookup. As part of the thesis research, control plane implementation of OpenLISP is done. Assuming transit /edge approach is widely deployed in the future Internet; our approach to live migration eliminates the need for non-standard and complex virtual networking schemes to interconnect the VM platforms. Our solution guarantees optimal routing between clients and the VM end-point that moved, regardless of its location. In addition, the solution does not require any change in the DNS infrastructure, which overall reduces operating expenses for the data center administrator.

Chapter 1: Introduction

1.1 Background

Cloud computing has emerged as a compelling paradigm for the deployment of distributed applications and services on the Internet due in large to the maturity and wide adoption of virtualization technologies. Virtual machines (VM) [1] are becoming increasingly valuable for resource consolidation and management, providing efficient and secure resource containers, along with desired application execution environments. Coupling those resource utilization benefits with VM migration [2] capabilities enable dynamic, elastic data centers, and eventually grant important energy saving to large Internet service providers. Global mobility is imperative for data center maintenance without downtime, disaster avoidance, data center migration and consolidation, data center expansion, and workload balancing across multiple sites. The scale of cloud computing is growing as business applications are increasingly being deployed across multiple global data centers. At the global Internet scale, one would need wide-area migration of virtual machines, services and infrastructure from one network to another (distant) network.

1.2 Problem statement and main contributions

In principle, wide-area migration uses the same concepts as legacy local migrations. However, two important factors impede an efficient deployment of wide-area migration across Internet Clouds. First, the bandwidth between source and destination site is likely to be severely limited in comparison to a local connection – i.e. the time needed for the state transfer will increase. Second, changing the location of the VM implies different addressing at the new location under the legacy IP/BGP (Internet Protocol/Border Gateway Protocol) [3] Internet infrastructure. In this thesis, we address the second issue. The big impediment is the lack of an adequate control-plane in the BGP architecture, which is the de facto inter-domain routing protocol on the Internet. Its protocol architecture is rather simple and offers flat routing with no hierarchy; its simplicity comes at the expense of performance, yielding often poor user's Quality of Experience, so important for Cloud services.

In this thesis, we examine the efficacy of live Virtual machine migration over the present Internet. We focus on what we consider to be the obstacle: moving the network state of the VM from one location to another without loss of service, while counterbalancing possible additional inefficiencies that BGP could introduce to support these operations. The significant growth of Internet traffic and increase of routing tables require solutions to address Internet scalability and resiliency. A possible direction is to move away from the flat legacy Internet routing to hierarchical routing, separating edge networks from transit networks [4]. A protocol, the Location Identifier Separation

Protocol (LISP) [5], under standardization at the IETF and under progressive deployment today, follows this direction. It adds new control-plane functionalities based on a mapping system, which maps the IP network location of a host/network to its IP identification, separating it hence adding a two-level Internet hierarchy routing. In LISP, host Endpoint IDs (EIDs) are numbered from a separate, non-provider- assigned and non-topologically-bound space, they do not need to be renumbered when a client site changes its attachment points to the network. In the context of Cloud computing and Internet datacenter, network-based Id/locator splitting provides a convenient method for establishing the network context for a VM that is migrated to a new physical machine.

An implementation of the LISP control-plane today exists only in some routers. During the thesis, work was performed for the implementation of control-plane modules for an OpenSource software router called OpenLISP[6], which will allow many networks worldwide to join the LISP-based Internet with very low cost. Based on FreeBSD [7], OpenLISP provides complete Data Plane support for both encapsulation and decapsulation operations for IPv4 and IPv6, as well as EID-to-RLOC (Routing Locator) mapping storage and efficient lookup. Another important contribution of this thesis is the definition, implementation and validation of the required procedures to achieve a live virtual machine migration across distant Internet datacenters, with particular emphasis on the Xen virtualization platform [8]. The solution is validated using real-life application servers migrating across data centers while clients accessing these applications. In addition, we present a set application-level benchmark aimed at evaluating important metrics of VM migration. The performance of different traffic types is evaluated assuming different distances among datacenters, as a function of the Cloud application type, bandwidth and distance between datacenters. The results are promising and open the path towards a transit-edge separation oriented internetworking protocols for the future Internet, avoiding non-standard patches to current BGP-based Internet and complex virtual networking schemes to interconnect Internet Clouds to the users.

1.3 Organization

The remainder of the thesis is organized as follows. Chapter 2 provides a survey of current mainstream Virtual Machine Migration techniques. Chapter 3 describes our proposed solution based on transit-edge separated Internet routing. Chapter 4 discusses the implementation issues and experimental setup of the Virtual Machine Migration based on transit-edge separation system. Chapter 5 discusses the rationale behind transit-edge separated Internet routing in details and analyses the experimental results. The thesis is concluded in Chapter 6.

Chapter 2: Review of Virtual Machine Migration Techniques

The growing use of virtualization in the data center has enabled an unparalleled degree of flexibility in managing servers and workloads. One important feature of this newfound flexibility is mobility. As workloads are hosted on virtual servers, they are decoupled from the physical infrastructure and become mobile.

As end-points become detached from the physical infrastructure and are mobile, the routing infrastructure is challenged to evolve from a topology centric addressing model to a more flexible architecture. This new architecture should allow IP addresses to freely and efficiently move across the infrastructure. There are several ways of adding mobility to the IP infrastructure, and each of them addresses the problem with different degrees of effectiveness. In this Chapter, we discuss the goals of IP mobility solutions. We also discuss the advantages and disadvantages of certain methods and the motivation of transit/edge separation system proposed in this thesis.

2.1 Goals of Virtual Machine Mobility

The requirements for a Virtual mobility solution can be generalized to a few key aspects. In order to do a fair comparison of existing solutions and clearly understand the added benefit of the proposed solution, we will discuss on the different aspects that must be addressed in a Virtual mobility solution.

Redirection towards the migrated machine:

The ultimate goal of VM mobility is to provide a solution that is totally transparent to the higher layer applications. It should re-direct the traffic to the valid location of the end-point after the migration and thereby, allow for the preservation of established sessions .Redirection can be achieved by replacing the destination address with a proxy address that is representative of the new location of the end-point. Different techniques will allow the redirection of traffic either by replacing the destination's address altogether or by leveraging a level of indirection in the addressing such as that achieved with tunnels and encapsulations. The different approaches impact applications to different degrees.

Routing Optimization:

As end-points move around, it is key that traffic is routed to these end-points following the best possible path. Since mobility is based largely on re-direction of traffic, the ability to provide an optimal path is largely a function of the location of the re-directing element. Depending on the architecture, the solution may generate sub-optimal traffic patterns often referred to as traffic triangulation; hair-pinning in an attempt to describe the unnecessary detour traffic needs to take when the destination is mobile. A good mobility solution is one that can provide optimized paths regardless of the location of the end-point.

Endpoint Independent Solution:

It is important that the solution is independent of agents installed on the virtual machine end-points or on the clients interacting with these end-points given the precedent of the large installed base of end-points that cannot be changed or managed at will in order to install client software. A desirable approach will be network based solution and is imperative to the scalable deployment of the solution.

Address Family Independent Solution:

The most important aspect of the solution is it must work indistinctly for IPv4 or IPv6 end-points and networks. Since mobility relies on the manipulation of the mapping of identity to location, exclusive solution for address families with lengthier addresses tend to be incompatible for the smaller address spaces. These addresses dependent solutions have limited scope in deployment as they usually call for an end to end deployment of only IPv4 or only IPv6. In order to broaden the effectiveness, the ideal solution should work for IPv4 or IPv6 indistinctly.

In the remainder of this Chapter, we will discuss typical examples of different Virtual Machine Mobility techniques.

2.2 Existing IP Mobility Solutions

Live migration techniques within a LAN segment were developed in Xen [9], and also in VMware was developed in [10] [11]. Xen demonstrated live migration of a session of an online game, Quake 3 in 60 ms [12]. Qin Li, et al. [13] proposed a solution which was based on Mobile-IP i.e., Virtual Machines were given Mobile IP addresses, and was shown that hypervisor is capable of predicting exact time and destination host of Virtual Machine migration. The downtime was about 30 sec. Mobile IPv4 solution suffers with triangular traffic pattern. Also, Mobile IPv4 does not extend the solution for multicast i.e. multicast traffic from the mobile node always has to hairpin through the home agent since the distribution tree is built and rooted at the home agent. Since the mobile node is usually sourcing traffic, if the Foreign Agent is not directly connected, there is a need to inject the host route at the foreign site to get RPF (Reverse Path Forwarding) to work.

Bradford, et al. [15] experimented on migrating Web servers through a LAN and a WAN using an approach based on dynamic DNS (Domain Name System) and tunnels in the WAN. The downtime was 3 sec in the LAN, and 68 sec in the WAN.

The main disadvantages with this approach are:

- If the rate of refresh is too high, then the DNS cache may impact either the convergence time for the move or the scalability of the DNS system.
- It doesn’t scale for IP based connection model as it works only for name - based connections.
- This solution suffers from hair-pinning i.e. that there is a period of time where there are active connections to the old address and also some new connection requests to the new address in the second Virtual Machine. There is a state of ambiguity for the network administrator during this state as they are unable to ascertain that these two addresses are the same system (from the application point of view).

A simpler method for wide-area migration is to use a layer 2 VPN, such as VPLS (Virtual Private LAN Service), to connect two data centers by Ethernet protocol. Again, in L2VPN based methods, packets from clients (VM users) to a VM are redirected by the source data center to the destination center. So the WAN path between the clients and the VM is not optimized and suffers from hair-pinning

In IETF RFC 3775, mobility support for IPv6 is discussed. The solution is similar to that of IPv4 with a few additions. Once the machine is migrated, the Home Agent is taken out of the data path by distributing the Care-of-Address (CoA) to Home Address Binding information to the client itself. So, the Home Agent doesn't redirect the traffic to the CoA for the server that has moved. Once the client has the CoA information for a particular server, it can send traffic directly to the CoA rather than triangulating it through the Home Agent. Although Mobile IPv6 provides direct path routing for mobile nodes, this solution is limited to IPv6 enabled end-points, and the primary requirement is the entire data path should be IPv6 enabled, and also it also requires that the end-points are enabled with IPv6 mobility agents.

We proposed to solve the Virtual Machine Migration problem based on Transit/Edge separation system of the Internet. The solution proposed in this thesis only need to be implemented on some selected edge routers of the Internet to create an additional overlay network. Thus, the expense of upgrading the network can be kept to a minimum. Meanwhile, the core of the internet still runs the legacy BGP protocol.
The solution will be discussed in Chapter 4 and 5.

2.3 Conclusion

In this Chapter we presented a discussion of current Virtual Machine Migration techniques. The advantages and limitations of various algorithms were considered. Based on the discussion given, none of those algorithms perfectly fits our requirements, as presented in Chapter 1. This motivates us to develop the transit/edge system to be discussed in Chapter 3 and 5.

Chapter 3: Transit Edge Separation Internet

The Internet has evolved from an academic network managed and operated by researchers, to a worldwide and ubiquitous network interconnecting devices of multiple natures .The Border Gateway Protocol (BGP) [3] is the only inter-domain routing protocol used by Autonomous Systems (AS) to exchange routing information. In recent years, due the growth of Internet, there is a fast increase of BGP routing table size in the default free zone (DFZ).This indicates a potential scaling challenge in the case of migrating today's Internet to IPv6 . IPv6 has a vastly enlarged address space compared to IPv4. Many solutions have been proposed address this scaling problem of the inter-domain routing. Among them, the transit-edge separation scheme looks more promising than others due to its practical advantages. In this chapter, we discuss the scalability and resiliency problems faced by the current Internet. Also, the evaluation on how the scalability can be improved from transit-edge separation scheme is done. Also, we introduce a new protocol based on transit-edge paradigm called LISP which we will be using in the Virtual Machine Migration testing in Chapter 5.

3.1 Analysis

Analyzing current transit BGP routing tables from Routeviews [16], growth of ASes from 2000 to present is plotted in Fig 3.1; we find that currently the Internet is composed of about 35,000 ASes which indicates the fast growth of Internet in recent years. This number doesn't stop here and is predicted to grow higher. Also we find that, ASes which appear at the last position account up to 84% in the BGP routing table. These ASes typically called Stub ASes represent Cloud/content providers, large corporations, universities. Analyzing the historical trend of AS stub number ratio, one can infer that it has been linearly increasing for the past few years.

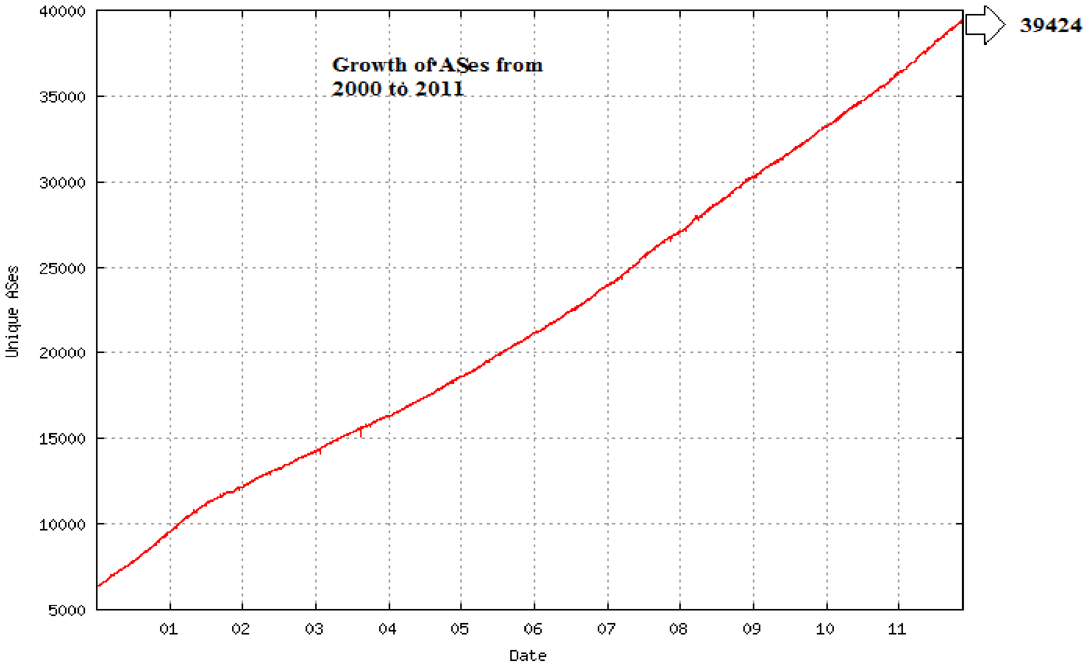


Figure 3. 1 Growth of ASes from 2000 to present

Tier-3 ASes include many large stub ASes that have fragmented their operational network into many dependent ASes, or small service providers offering Internet services in small geographical regions. They account for 10% in the BGP routing table. Finally, large tier-3s, those appearing at most in the third from last position are about 3%. Stub and tier-3 ASes thus represent the large majority, about 97%, and can be considered the *edge* of the Internet. Most of them are multi-homed, i.e., they have more than one upstream provider connecting them to the rest of the Internet, and about 17% of them are connected to more than two providers.

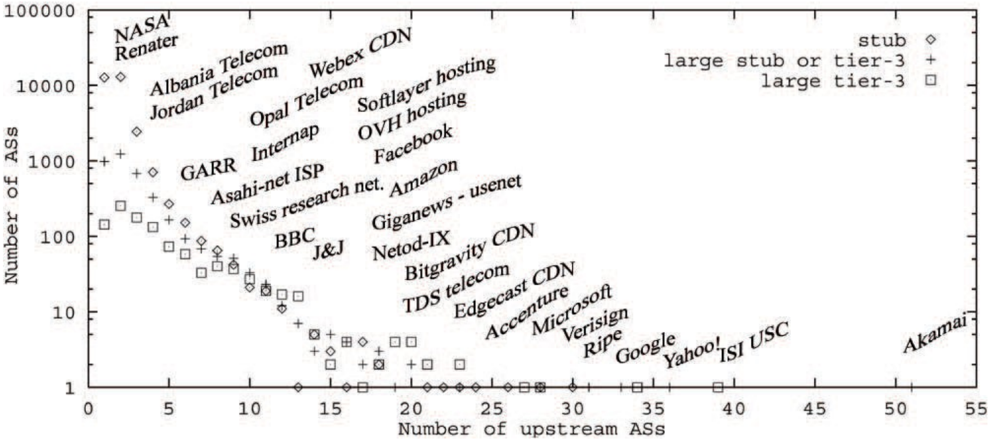


Figure 3. 2 Multi-homing distribution of destination ASes [4]

Fig. 3.2 shows the distribution of the number of upstream ASes per stub AS, large stub or tier-3 ASes (at most penultimate position in AS paths), and large tier-3s .ASes with high degree of multi-homing are Cloud/content providers (e.g., Amazon, Google) and content delivery networks (e.g., Akamai, Edgecast), while service providers (e.g., Verisign, Internap) or research networks (e.g., Renater) are multi-homed to few providers.

Multihoming brings in an advantage of traffic engineering and network reliability benefit from an improved interconnectivity. Here, the term traffic engineering consists of controlling the direction and the load of inbound and outbound traffic from and towards the upstream ASes.

At present the traffic engineering is performed in the legacy BGP protocol in two ways. One is through its attribute, the local preference. The local preference can be assigned to incoming BGP messages to rank the upstream networks. Another way to perform traffic engineering is through a method, the AS path prepending. In AS path prepending one can artificially increase the AS path to distract traffic volumes toward its other providers. Local preferences cannot precisely be inferred looking at routing tables, while one can notice prepended AS paths. We find that about 17.5% of the edge AS networks are actively using the path prepending, with at least 2 upstream ASes. These edge AS networks have thus strict Internet traffic engineering requirements for their services. Nevertheless, this has been effective, the Internet traffic engineering resulting from BGP attribute tweaking remains incomplete, time-consuming and highly computational intensive for routers. It also results in an excessive fragmentation of network prefixes that is exploding the BGP routing table size. About 30% of edges AS networks announce more than 100 network prefixes. Recent detailed analysis shows that the size of the routing table can be reduced by 43% to 90% at different levels of transit-edge routing separation [17]

A feasible direction is to address in a scalable way the separation between the transit and the edge routing domains. With transit-edge separation, the edge-to-edge routing decision is enriched: not only the best path toward the destination edge network has to be chosen, but also the best locator and/or the best egress gateway for the source edge network. Transit-edge routing separation allows important performance enhancements - such as a significant reduction of the routing table size, seamless mobility management, Internet routing security preservation. In the current internet architecture, a single namespace, namely the IP address, is used for both indentifying and locating end-systems.

In the transit-edge separation architecture, Locator/ID Split paradigm is employed with two different namespaces used to respectively identify and locate end-systems. Different protocols can be used to manage the identifier to-locator mappings and to encapsulate the packets in the transit sub-path, such as the Locator-Identifier separation protocol (LISP) [5] which is currently under standardization. There are other host based approaches such as SHIM6 [18] or HIP [19] that appear less scalable. Such a separation aims at solving the scalability issues that the current Internet is facing, mainly concerning the continuously increasing BGP routing table, but also concerning addressing, mobility, multi-homing, and inter-domain traffic engineering.

3.2 LISP

The Locator Identity Separation Protocol (LISP) is a new routing architecture that creates a new paradigm by splitting the device identity, known as an Endpoint Identifier (EID), and its location, known as its Routing Locator (RLOC), into two different numbering spaces. This capability brings renewed scale and flexibility to the network in a single protocol, enabling the areas of mobility, scalability and security.

3.3 Definition of Terms

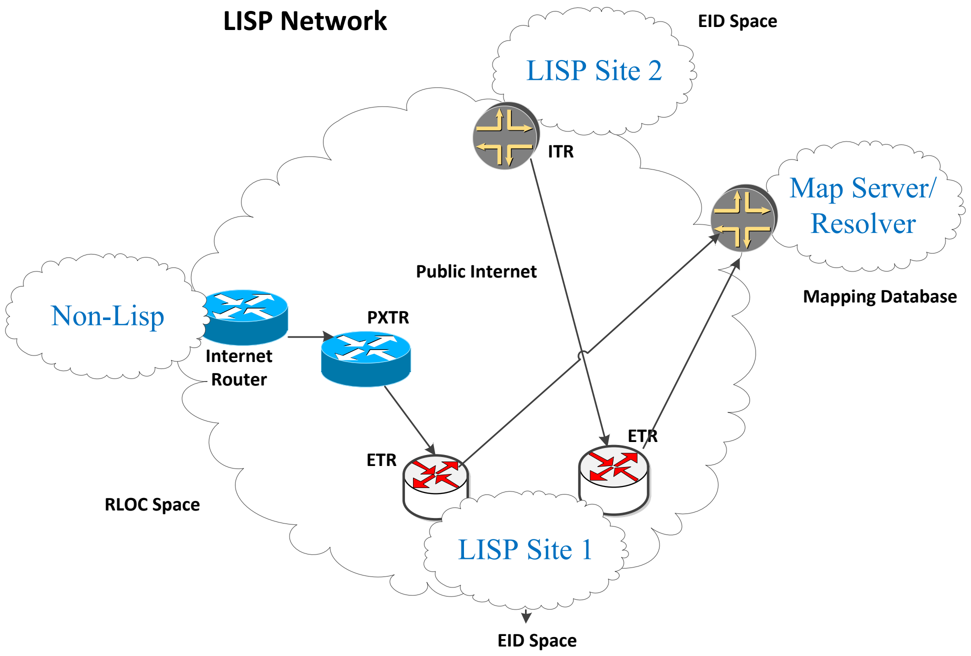


Figure 3. 3 LISP Network Elements

A LISP-enabled network includes some or all of the following components:

Routing Locator (RLOC): Consists of the IP addresses and prefixes identifying the different routers in the IP network. Reachability within the RLOC space is achieved by traditional routing methods**.** Typically, RLOCs are numbered from topologically-aggregatable blocks that are assigned to a site at each point to which it attaches to the global Internet; where the topology is defined by the connectivity of provider networks, RLOCs can be thought of as provider aggregatable (PA) addresses.

Endpoint ID (EID): Consists of the IP addresses and prefixes identifying the end-points.

The host obtains a destination EID the same way it obtains a destination address today, for example through a Domain Name System (DNS) [21] lookup or Session Invitation Protocol (SIP) [22] exchange. An EID is allocated to a host from an EID-prefix block associated with the site where the host is located. EID reachability across LISP sites is achieved by resolving EID-to-RLOC mappings.

Ingress Tunnel Router (ITR): An ITR is a LISP Site edge router that receives packets from site-facing interfaces (internal hosts) and encapsulates them to remote LISP sites, or natively forwards them to non-LISP sites.

Egress Tunnel Router (ETR): An ETR is a LISP Site edge device that receives packets from core-facing interfaces (the Internet) and decapsulates LISP packets and delivers them to local EIDs at the site. ETR functionality does not have to be limited to a router device. A server host can be the endpoint of a LISP tunnel as well.

LISP Infrastructure Devices:

Map-Server (MS): An MS is a LISP Infrastructure device that LISP site ETRs register to with their EID prefixes. The MS advertises aggregates for the registered EID prefixes into the LISP mapping system. All LISP sites use the LISP mapping system to resolve EID-to-RLOC mappings.

Map-Resolver (MR): An MR is a LISP Infrastructure device to which LISP site ITRs send LISP Map-Request queries when resolving EID-to-RLOC mappings.

Proxy ITR (PITR): In order to provide interoperability with non-LISP sites, Proxy ITR/ETR devices, abbreviated PxTR, are deployed within the Internet. A PITR is a LISP Infrastructure device that provides connectivity between non-LISP sites and LISP sites by attracting non-LISP traffic destined to LISP sites and encapsulating this traffic to LISP sites.

Proxy ETR (PETR): A PETR is a LISP Infrastructure device that allows IPv6 LISP sites that have only IPv4 RLOC connectivity to reach LISP and non-LISP sites that have only IPv6 RLOC connectivity.

EID namespace is used within the LISP sites for end-site addressing for hosts and routers. These EID addresses go in DNS records, just as they do today. Generally, EID namespace is not globally routed in the underlying Internet. RLOC namespace, however, is used in the (Internet) core. RLOCs are used as infrastructure addresses for LISP routers and core (service provider) routers, and are globally routed in the underlying infrastructure, just as they are today. Hosts do not know about RLOCs, and RLOCs do not know about hosts.

3.4 LISP performance evaluation in Cloud Computing

In order to examine how LISP improves the performance for cloud computing, this section provides details of design and verification phase with two cases. In the first case, analysis will be done in non-LISP environment and the second case, in LISP environment.The topologies are built in GNS3[22]

At Cloud Provider, we configure the video stream server with RSTP protocol at port 5544 for controlling the video streaming. At client side, we open network stream using a video streaming client.

Video streaming runs over UDP. UDP is a best-effort protocol, which do not take care of retransmission. We identify the number of packet lost in video streaming during the change in the network both in LISP and in native BGP scenarios, by analyzing the media packets captured in the client location .Fig 3.4 shows the steady state video streaming seen from the client side.

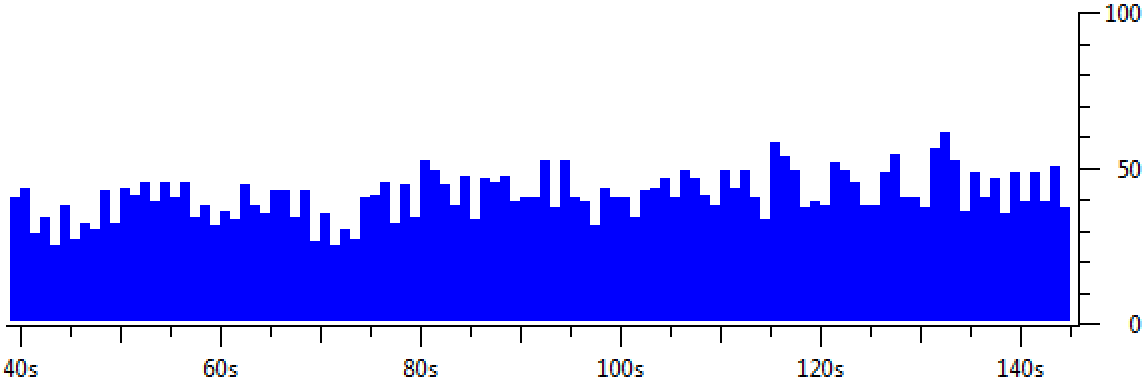


Figure 3. 4 Steady state video streaming before locator change

3.4.1 Scenario for Non-LISP environment

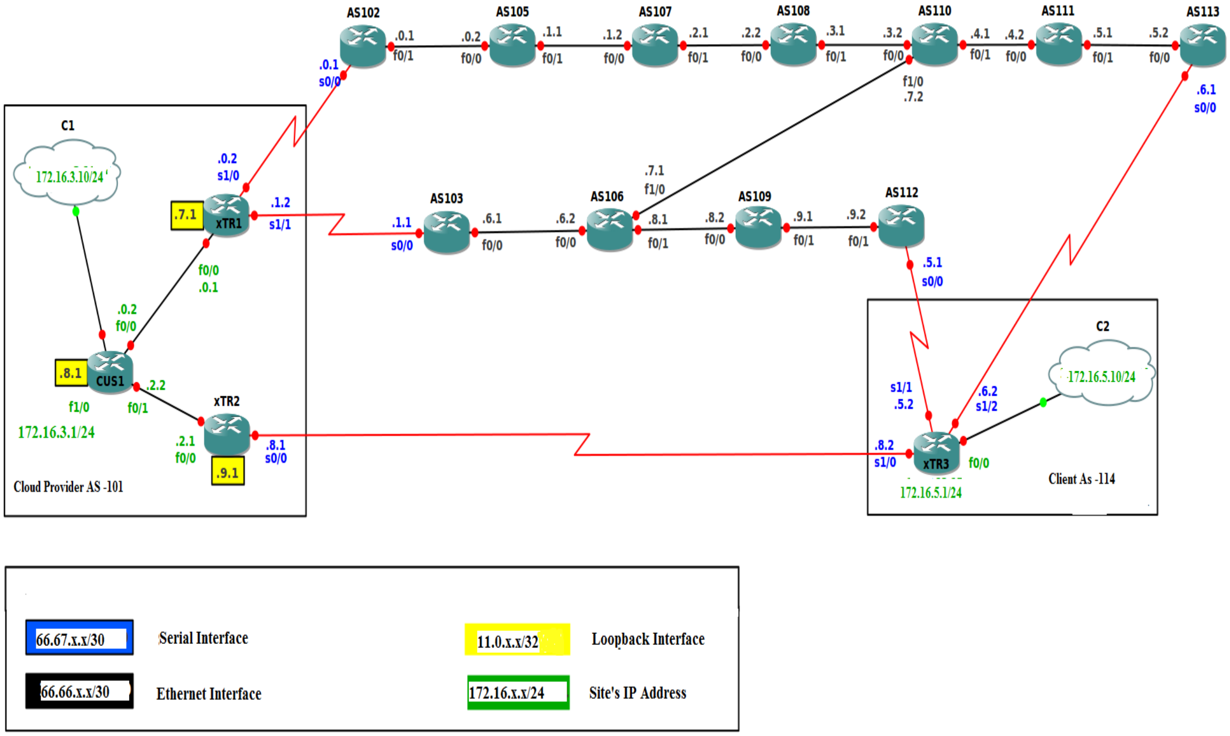


Figure 3. 5 Topology for Non-LISP scenario

Fig 3.5 shows the topology for the BGP scenario. In this scenario, each site is multi-homed, the topology consists of fifteen routers and all of these routers will provide the functionalities in non-LISP environment. Eleven routers will be part of Internet Service Providers, the others operate at Cloud Provider's site and Client's sites. BGP (Border Gateway Protocol) is the protocol used to exchange routing information across the whole topology. Each edge router of Cloud provider site is multihomed to one or many gateways of ISP. Client also has many different paths (both long and short path) to reach the Cloud provider. If any route in the Cloud Provider site is changed, all other routers have to update their routing table accordingly .

Changing locator

To change the locator on BGP, withdraw route is used with the following step:

1. At first, let Client access Cloud provider through xTR2 (the shortest way).
2. At router Cus1 , the advertisement of the network 172.16.3.0/24 is not distributed to xTR2 so that the incoming traffic go through xTR1.

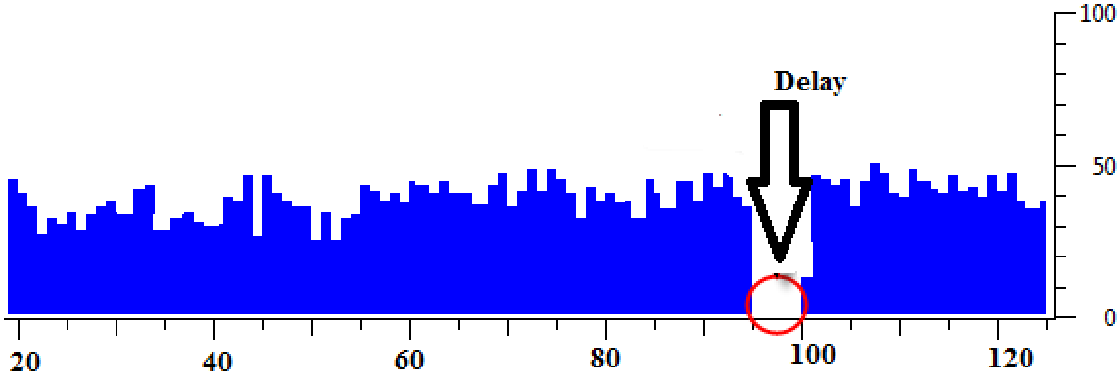


Figure 3. 6 Packet Loss after changing locator for Non-LISP scenario

For the current network, BGP routing has been used widely by the ISPs because it has many advantages in routing path between ASes. We measure the jitter which is defined in RFC 3393 as IPDV (Instantaneous Packet Delay Variation). In order to see how IPDV affect the video quality, routing path from Cloud Provider to Client is changed to cause latency for the packets which lead to delay situation.

In this demo, when a network change was made, it took a long time for BGP to adapt the new route, which affected the packet flow in the network. As a result, there was considerable amount of packet loss as seen in Fig 3.6. This is one of the disadvantages when comparing with LISP.

3.4.2 Scenario for LISP environment

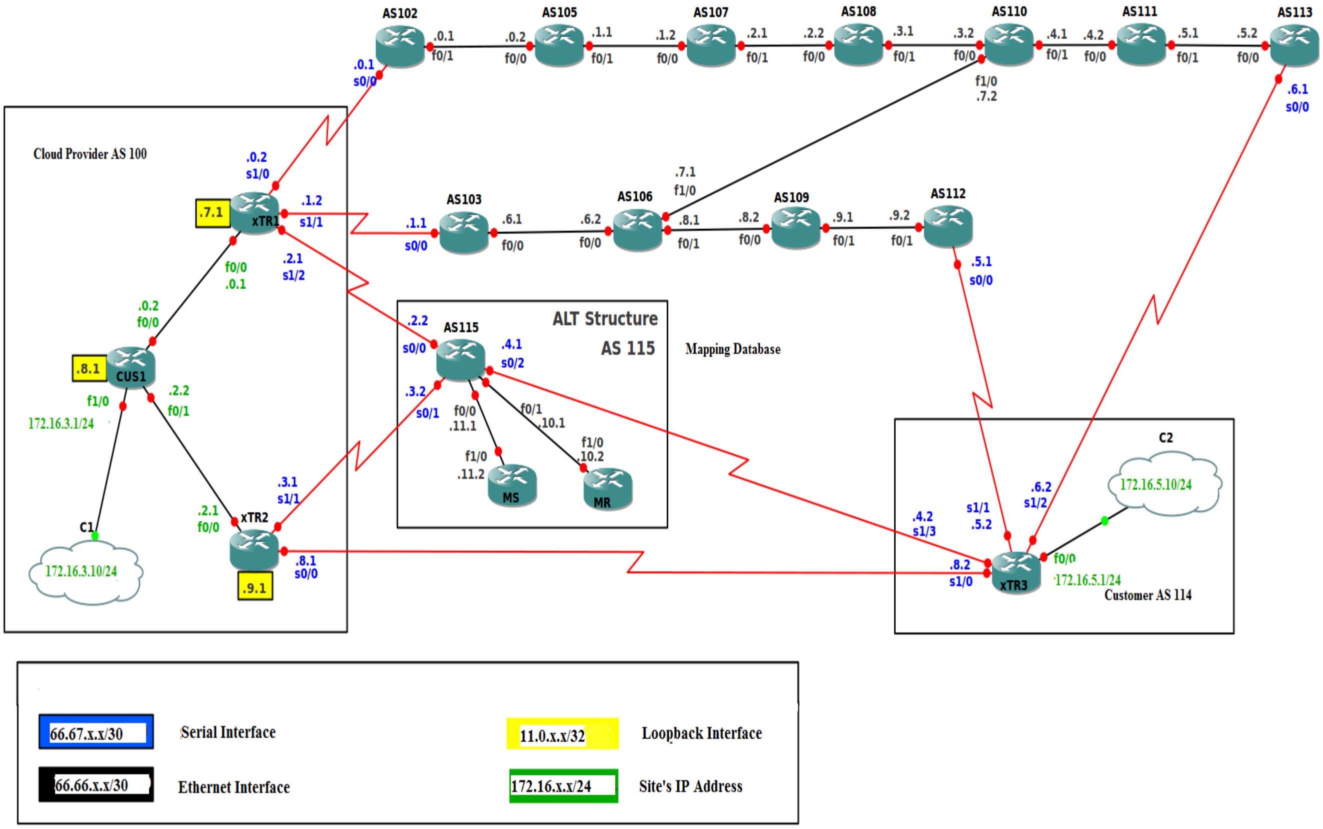


Figure 3. 7 Topology for LISP Scenario

This topology requires eighteen routers, including five LISP capable routers. The Cloud provider and Client site are both multi-homed. The edge routers of each site use Cisco enabled LISP IOS and act as Ingress Tunnel Routers/Egress Tunnel Routers (abbreviate as xTRs). Each site has three Routing locators (RLOCs) corresponding to two different Service Providers and one directly connected with xTR of the other site. The two other routers function as a Map Server and Map Resolver (MS and MR). The remained routers operate as the non – LISP core.

To change the locator on LISP, the priority of the RLOCs is changed with the following steps:

1. At first, let Client access Cloud provider through xTR2 (the shortest way).
2. At router xTR1 and xTR2 , they update the network 172.16.3.0/24 to Map Server, and then the xTR3 know that how it can reach 172.16.3.0/24

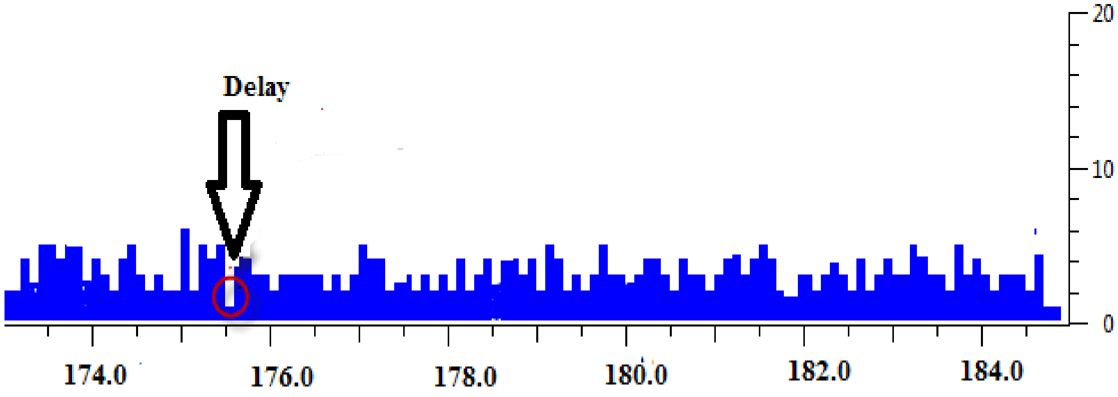


Figure 3. 8 Delay in Video Streaming in LISP scenario

After the changing the locator of Cloud Provider site, the edge routers compute new path quickly in less than a second, so the delay time is reduced greatly as seen in Fig 3.8. The video quality is not deteriorated while the change of locators happen .The following table shows the summary of the comparison between two simulations for the video application:

Criteria	**BGP**	**LISP**
Rerouting delay	~9s	~0.06s
Packet loss	~275/5600	No loss
Video quality	Pause during rerouting	Played continuously

In BGP topology, when Cloud provider changes its locators, all the routers have to recalculate the routing table, so the routers drop packets for which they do not have a valid next hop, or queue packets while awaiting the completion of routing table. While in LISP, all the routers do not have to recalculate the routing table. It only takes a little time for the edge router of each site to get the mapping information through the LISP infrastructure. In both simulations, the packet loss only happens during the time of rerouting.

3.5 Analysis and conclusions

The legacy flat-routing approach to Internet routing, under which the source network decides the AS path directly to the destination network, is showing all its deficiencies in terms of scalability and resiliency. Placing intermediate gateways and locators separating edge networks from transit carrier networks can allows important performance improvement. In this Chapter, we study the novel traffic engineering capabilities arising in a transit-edge routing separation context.Also in this Chapter, we analyzed the delay behavior in LISP and BGP in the Cloud Computing scenario. It was established that changes in the routing locator in the LISP system is reflected in mimimal delay in network convergence.The experimental results discussed in this Chapter form the basis for the Virtual Machine Migration technique developed in Chapter 5.

Chapter 4: Implementation Issues and Test-Setup

LISP is a Locator/Identifier separation solution based on the transit-edge internet system as described in Chapter 3. In this chapter, we present OpenLISP, an open source implementation of the LISP Protocol running in the kernel of the FreeBSD Operating System. Natively, OpenLISP focuses on the data plane operation, implementing the LISP-Cache and the LISP-Database in the kernel space, as well as the encapsulation/decapsulation functions. Control plane is originally not developed. As part of the thesis, the control plane for the OpenLISP was developed and tested in the LISP Beta Network testbed. This Chapter also describes the messages of the LISP protocol and the building blocks of OpenLISP framework. Implementation issues and the experimental setup for the LISP Beta Network are also discussed.

4.1 OpenLISP

4.1.1 OpenLISP Data Plane

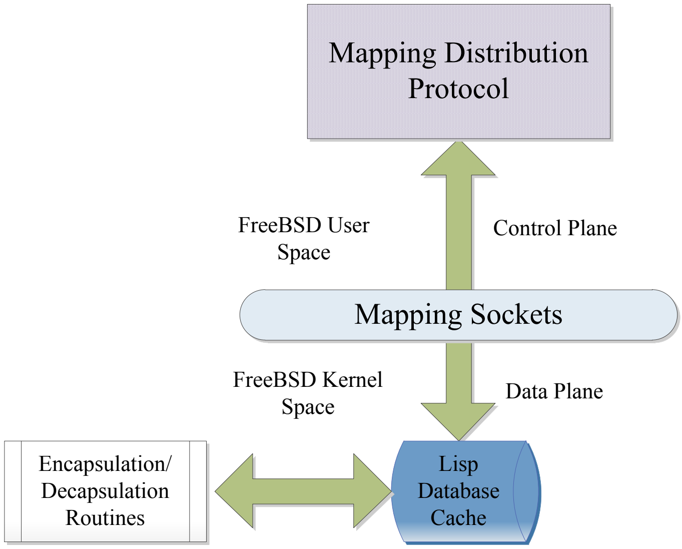
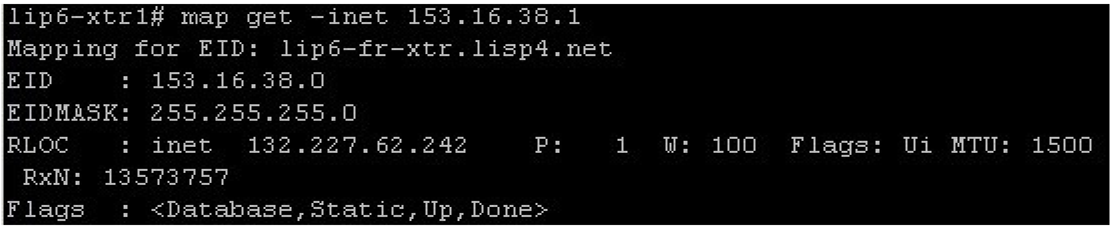


Figure 4. 1 OpenLISP Framework

OpenLISP is implementation of LISP in the FreeBSD [7] operating system. The high-level architecture of OpenLISP is depicted in Fig.4.1. LISP Data Plane is implemented directly in the kernel space, which contain functions to perform encapsulation and decapsulation as well as both LISP's cache and database, both merged in a single data structure called MapTable.

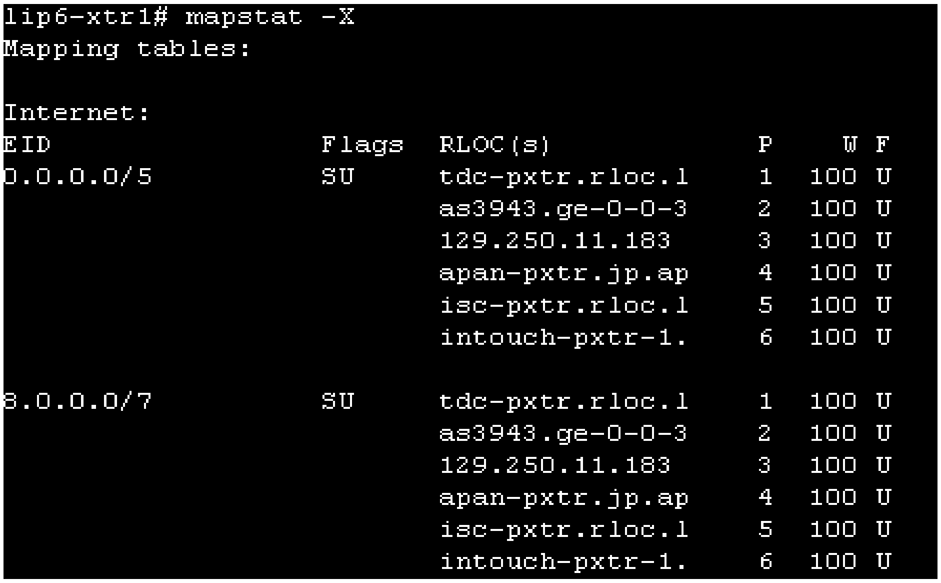
OpenLISP Data Plane is accessed through two tools , namely map and mapstat .The map utility has similar functionalities to the route utility, present in UNIX systems, to provide a command-line interface to configure the MapTables.



```
lip6-xtr1# map get -inet 153.16.38.1
Mapping for EID: lip6-fr-xtr.lisp4.net
EID     : 153.16.38.0
EIDMASK: 255.255.255.0
RLOC    : inet  132.227.62.242    P:   1  W: 100  Flags: Ui MTU: 1500
 RxN: 13573757
Flags   : <Database,Static,Up,Done>
```

Figure 4. 2 map command

The map utility supports several general options and commands, enabling the user to specify any arbitrary request that could be delivered via the API described in the next section. Fig. 4.2 shows an example of usage of the map tool to perform a lookup for the EID 153.16.38.1.



```
lip6-xtr1# mapstat -X
Mapping tables:

Internet:
EID                 Flags  RLOC(s)            P    W F
0.0.0.0/5           SU     tdc-pxtr.rloc.l    1  100 U
                           as3943.ge-0-0-3    2  100 U
                           129.250.11.183     3  100 U
                           apan-pxtr.jp.ap    4  100 U
                           isc-pxtr.rloc.l    5  100 U
                           intouch-pxtr-1.    6  100 U

8.0.0.0/7           SU     tdc-pxtr.rloc.l    1  100 U
                           as3943.ge-0-0-3    2  100 U
                           129.250.11.183     3  100 U
                           apan-pxtr.jp.ap    4  100 U
                           isc-pxtr.rloc.l    5  100 U
                           intouch-pxtr-1.    6  100 U
```

Figure 4. 3 mapstat command

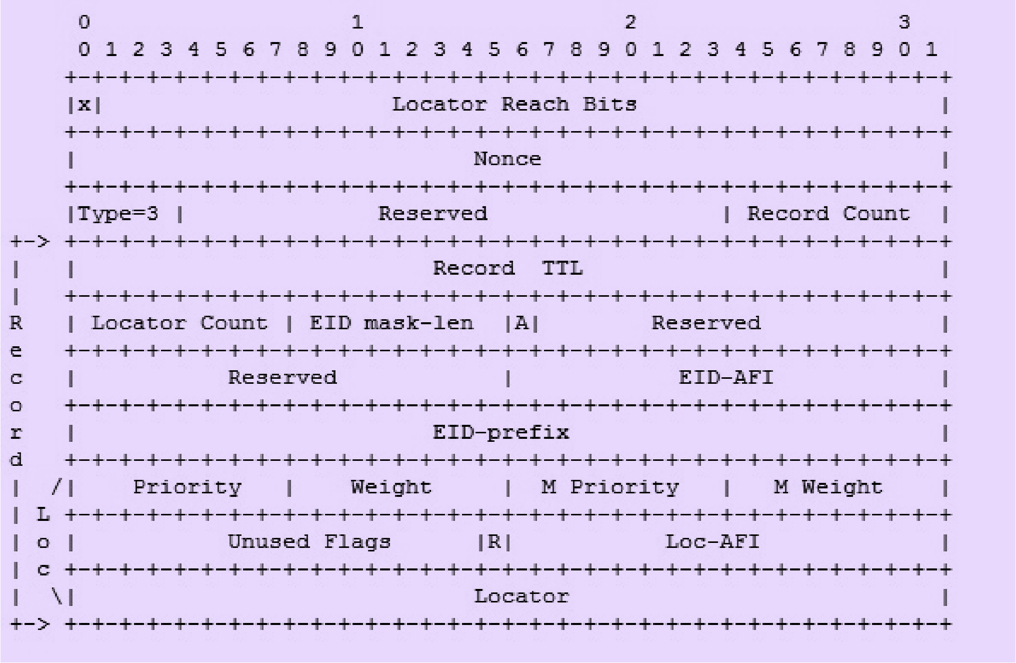
The mapstat command allows retrieving and displaying various contents of network-related LISP data structures. It is similar to the existing netstat command, hence, offering similar features but pertaining to the OpenLISP Data Plane. For instance, it is able to show complete information of the content of the MapTables and also statistics relevant to encapsulation and decapsulation operations. Fig. 4.3 shows an example of usage of the mapstat tool to display the contents of Map-Table.

The LISP specification defines both an encapsulation format for transporting data between hosts and a set of protocol messages that are used by the LISP infrastructure for managing EID-to-RLOC mapping information. Both data and control messages use User Datagram Protocol (UDP) transport to facilitate passage through the Internet. Encapsulated user data packets are transported using UDP port 4341, and LISP control packets are transported using UDP port 4342.

4.1.2 Control Plane Implementation

The control plane messages implemented are the following

Map Register

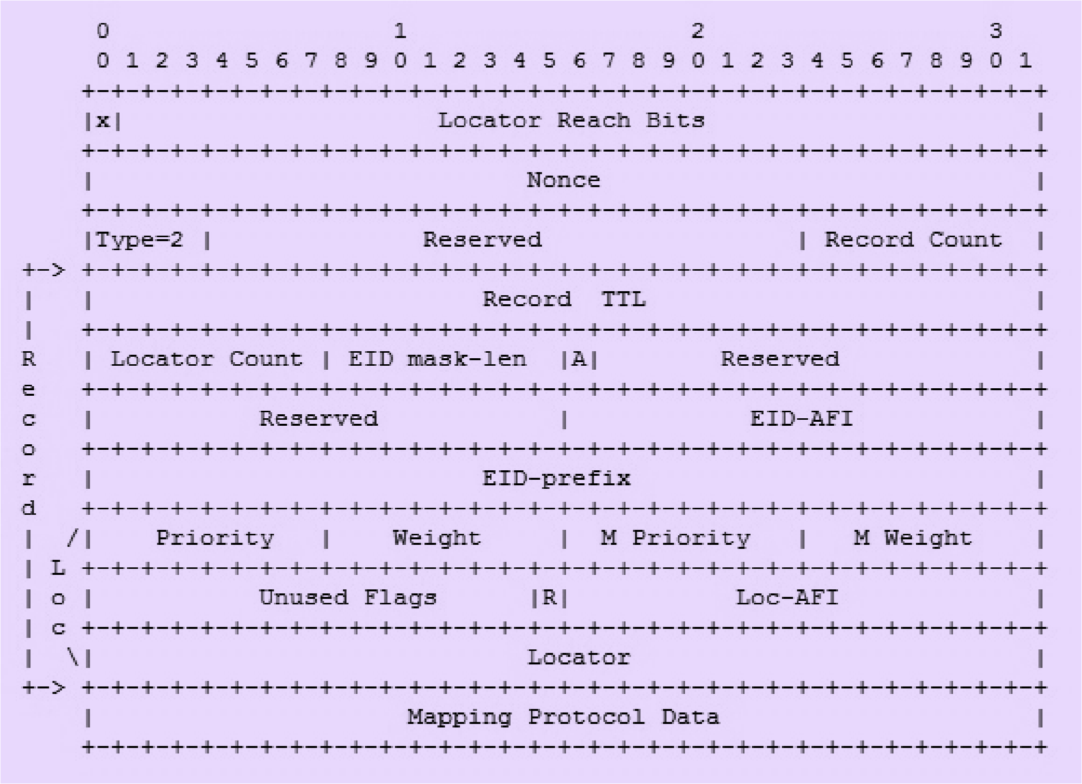


```
        0                   1                   2                   3
        0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |x|                    Locator Reach Bits                       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |                             Nonce                             |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |Type=3 |             Reserved                  | Record Count  |
   +-> +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |   |                          Record  TTL                          |
   |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   R   | Locator Count | EID mask-len  |A|          Reserved           |
   e   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   c   |           Reserved            |            EID-AFI            |
   o   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   r   |                          EID-prefix                           |
   d   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  /|    Priority   |    Weight     |  M Priority   |   M Weight    |
   | L +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | o |           Unused Flags        |R|           Loc-AFI           |
   | c +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  \|                            Locator                            |
   +-> +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4. 4 Map Register Message Format [5]

This message is sent by an ETR to a map server to define an EID prefix that it owns as well as the RLOCs that is used for exchanging Map-Request and Map-Reply messages. A Map-Register message includes authentication data The ETR and Map-Server is configured with a secret shared-key, before Map-Register message is sent. A Map-Server's configuration should also include list of the EID- prefixes for which each ETR is authoritative and should verify that a Map-Register received from an ETR only contain EID-prefixes that are associated with that ETR. The format of the Map-Reply is show above in Fig 4.4. Detailed explanations of all of the fields of Map-Register message are present in [5].

Map Reply



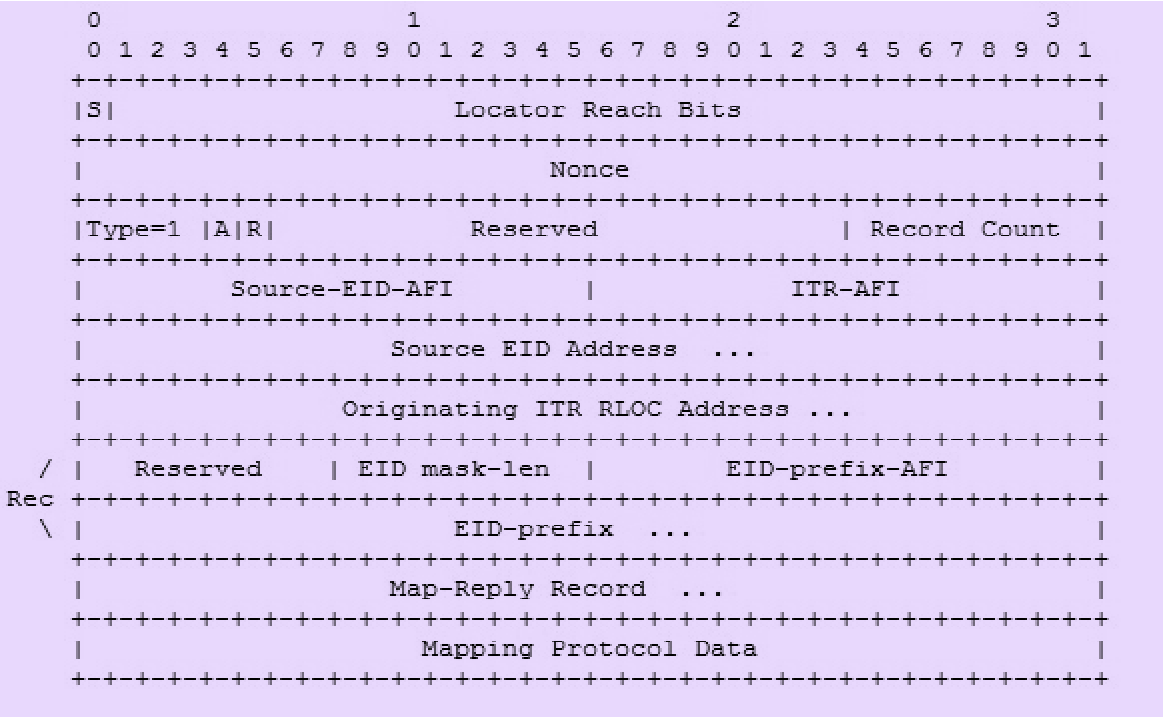
```
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |x|                    Locator Reach Bits                       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                            Nonce                              |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |Type=2 |              Reserved                 | Record Count  |
 +-> +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |   |                          Record  TTL                          |
 |   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 R   | Locator Count | EID mask-len  |A|        Reserved             |
 e   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 c   |            Reserved           |            EID-AFI            |
 o   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 r   |                          EID-prefix                           |
 d   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  /|    Priority   |    Weight     |  M Priority   |   M Weight    |
 | L +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 | o |            Unused Flags     |R|           Loc-AFI             |
 | c +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |  \|                            Locator                            |
 +-> +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |                     Mapping Protocol Data                     |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4. 5 Map-Reply [5]

This message is returned to an ITR by an ETR or map server in response to a Map-Request message .A Map-Reply message contains the EID prefix that matches the requested destination EID along with a set of RLOCs that can be used as the destination IP addresses for encapsulating user data. . The RLOCs present in the Map-Reply are globally-routable IP addresses of all ETRs for that LISP site. The End point Id being requested is either from the destination field of an IP header of a Data-Probe or the EID record of a Map-Request. The format of the Map-Reply is show above in Fig 4.5

The destination address is copied from the one of the ITR-RLOC fields from the Map-Request while sending a Map-Reply message. The ETR can choose a locator address from one of the address families it supports. The destination port of a Map-Reply message is copied from the source port of the Map-Request and the source port of the Map-Reply message is set to the well-known UDP port 4342. Detailed explanations of all of the fields of Map-Reply message are present in [5].

Map Request



```
     0                   1                   2                   3
     0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |S|                    Locator Reach Bits                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                             Nonce                             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |Type=1 |A|R|             Reserved              | Record Count  |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |         Source-EID-AFI        |            ITR-AFI            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                   Source EID Address  ...                     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                Originating ITR RLOC Address ...               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  / |   Reserved    | EID mask-len  |        EID-prefix-AFI         |
Rec +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  \ |                       EID-prefix  ...                         |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                   Map-Reply Record  ...                       |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                     Mapping Protocol Data                     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4. 6 Map-Request [5]

A Map-Request is sent by an ITR to the mapping database when it needs to send a packet to a destination EID for which is has no cached RLOC. There are three scenarios i.e. map requests are sent by an ITR when it needs a mapping for an EID, wants to test an RLOC for reachability, or wants to refresh a mapping before TTL expiration.

The format of the Map-Request is show above in Fig 4.6.

For the initial case, the destination IP address used for the Map-Request is the destination-EID from the packet which had a mapping cache lookup failure. For the latter two cases, the destination IP address used for the Map-Request is one of the RLOC addresses from the locator-set of the map cache entry. The source address is either an IPv4 or IPv6 RLOC address depending if the Map-Request is using an IPv4 versus IPv6 header, respectively.

In all cases, the UDP source port number for the Map-Request message is an ITR/PITR selected 16-bit value and the UDP destination port number is set to the well-known destination port number 4342. A successful Map-Reply, which is one that has a nonce that matches an outstanding Map-Request nonce, will update the cached set of RLOCs associated with the EID prefix range. Detailed explanations of all of the fields of Map-Request message are present in [5].

The implementation detail of the OpenLISP control plane is explained in Appendix A.

4.2 LISP4 Testbed

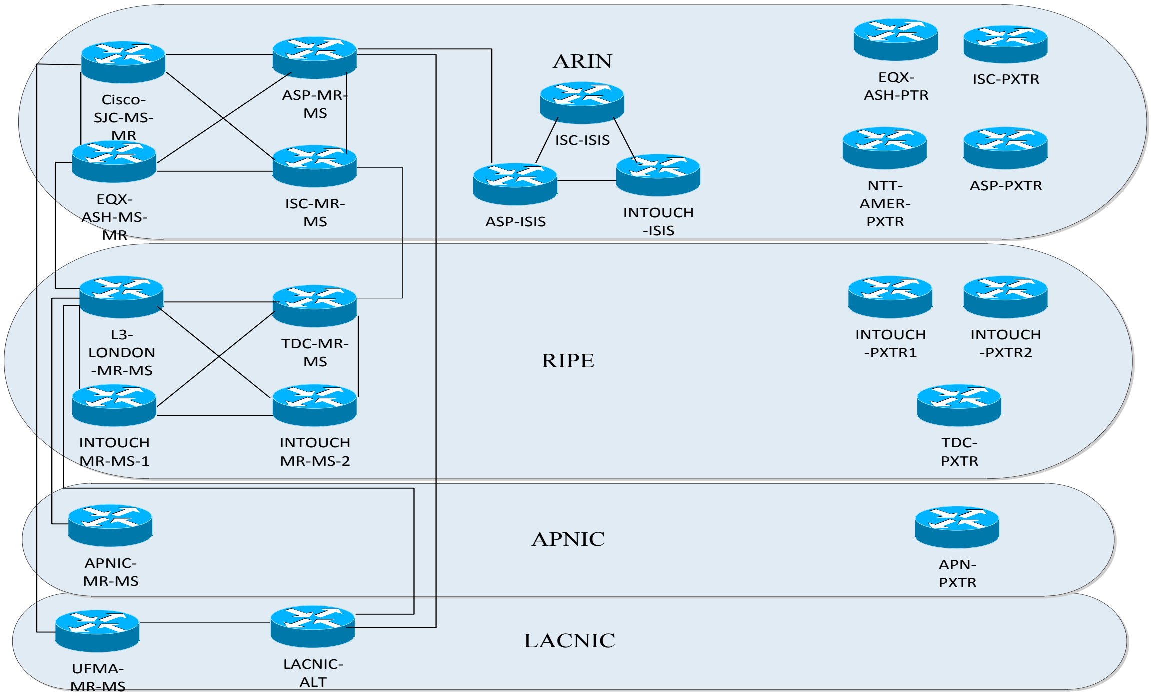


Figure 4. 7 LISP4 Beta Network

The LISP4 test bed is developed to gain real-life experience with LISP. At present, more than 100 companies from 20 countries are involved, including well known Internet Service Providers and Networking vendors like Google, Facebook, NTT, Level3, and the Internet Systems Consortium. For our experiments, we will be connecting to the LISP Beta Network to communicate with the other EID sites and also Non-EID (native Internet).

4.3 Tools used for the testing

4.3.1 Xen

The Xen hypervisor [22] is an open source software package that creates and manages partitions for multiple virtual machines. Partitions in a Xen environment are known as domains, with the management domain referred to as Domain0. Xen uses a virtualization architecture called para-virtualization, in which the guest operating systems are designed to take advantage of the fact—that they are running in a virtualized environment. With paravirtualization, the guest OS is modified to make special calls (hyper calls) to the hypervisor for privileged operations, instead of the regular system calls in a traditional unmodified OS. The applications in the guest OS's domain, however, remain unmodified. Our Virtual Machine Migration tests are done on Xen hypervisor 4.1.1.

4.3.2 netem

Netem [23] provides Network Emulation functionality for testing protocols by emulating the properties of wide area networks. It allows you to setup a delay profile on the outgoing packets on a network interface.

Some examples of the usage of the netem command are given below.

```
tc qdisc add dev eth1 root netem delay 60ms
```

This will add a fixed amount of 60ms of delay to all packets going out of the local Ethernet.

Random packet loss is specified in the 'tc' command in percent.

```
tc qdisc change dev eth0 root netem delay 40ms loss 0.1%
```

This will add 40ms of delay to all packets leaving the eth0 interface, along with the delay this causes 1/10th of percent (i.e. 1 out of 1000) packets to be randomly dropped.

Random noise can be emulated with the corrupt option. This introduces a single bit error at a random offset in the packet.

```
tc qdisc change dev eth0 root netem corrupt .1%
```

We use this tool to induce variable delays between the two networks in the Virtual Machine Migtration experiments to be shown in Chapter 5. This allows us to simulate the interaction of virtual machine hosts over a long-distance connection.

4.3.3 GNS3

GNS3 is a graphical network simulator that allows emulation of complex networks. It can be used to experiment features of Cisco IOS [24], Juniper JunOS [25] or to check configurations that need to be deployed later on real routers. Some of the features include simulation of simple Ethernet, ATM and Frame Relay switches. It has also the mechanism to connect the simulated network to the real world. The network topology used in the section "*LISP performance evaluation in the cloud computing*" which was discussed in Chapter 3 was built using this tool. Sample configuration of a network topology is shown in Appendix A.

4.3.4 VLC

VLC media player [26] is a highly portable, cross-platform multimedia player for various audio and video formats as well as DVDs, VCDs, and various streaming protocols. However in recent years it has also become an extremely powerful server to stream live and on demand video in several formats to our network and the Internet. Streaming video has become the dominant traffic type in the Internet. We will be using the VLC for live video streaming from the Cloud provider to the customer through the internet and analyze the streaming during the wide area virtual machine migration which will be discussed in Chapter 5.

Chapter 5: Virtual Machine Migration

In this Chapter, we discuss the details of the test set up for the Virtual Machine Migration in the LISP Network. Based on the OpenLISP implementation discussed in Chapter 4, we simulate the Cloud Computing scenario with different Service as a Service (SaaS) applications. The test set up for the Virtual Machine Migration is discussed and the results are analyzed. The process of migrations of virtual machines (or processes) would add scalability to virtualization solutions since virtual machines could be moved freely around different sets of hardware depending on bandwidth, available CPU cycles and latency requirements, while optimizing costs for the service provider.

5.1 Virtual Machine Migration Testbed

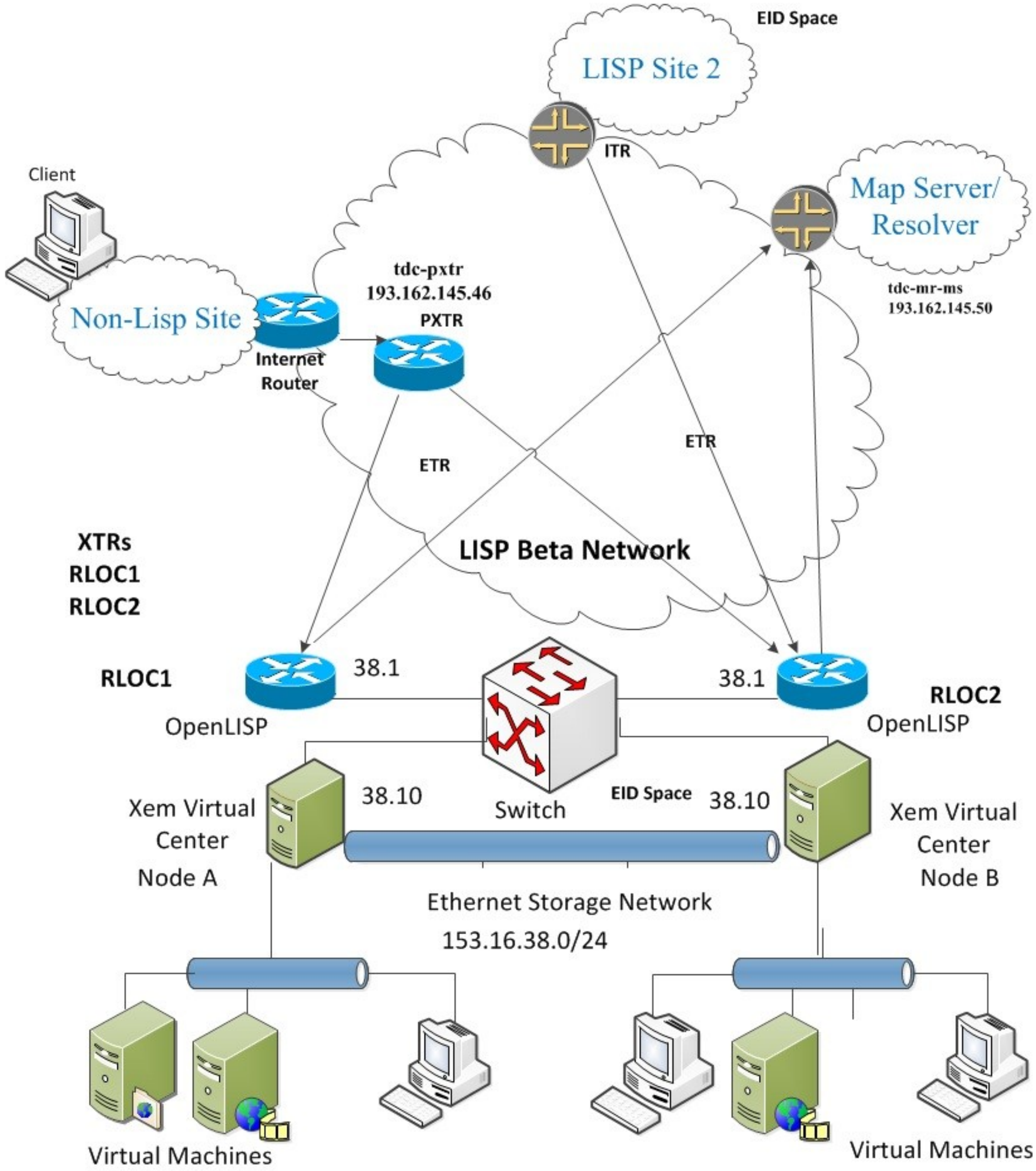


Figure 5. 1 Virtual Machine Migration Testbed

Virtual Machine Migration experiments are conducted in the LISP testbed. The LISP network is installed with OpenLISP nodes which interconnects to the LISP4 testbed mapping servers and the Proxy ITR.Our control plane impementation of OpenLISP are tested in these OpenLISP nodes.

The Data Center site is created in the LISP site as show in Fig 5.1. The two nodes correspond to the two Data Center sites. Our experiments are performed on identical hosts configured with an Intel Atom N 270 processor and 4 GB DDR RAM.Each host has an Intel Pro/1000 Gbit/s NIC to transfer the state of the VMs. The guest OS is Debian Linux with kernel 2.6.32.40 .All the VMs are configured to use 180 MB of RAM. The VMs use a 4 GB ext3 disk image as the standard basic image, which is installed with all the required libraries and dependencies .

When migrating a virtual machine to another host across the WAN, the two servers on separate networks may not have direct access to the same NFS server or cluster file system. Even if they did have access to the same NFS server, the latencies introduced by the virtual disk transfer over a WAN such as the Internet are likely to be significant. The amount of downtime an application will experience in a scenario where both Xen hosts have connectivity to the VM image is measured. The network delay is the total delay between the two Xen hosts. To generate the network delay, half of the desired delay is applied to the interface on VM Node 1 on VLAN 1, and the other half is applied to VM Node 2 interface on VLAN2. Multiple network delays are tested, which replicate different possible scenarios that would be experienced on the Internet. We use Linux's netem [25] to induce variable delays between the two networks. This allows us to simulate the interaction of virtual machine hosts over a long-distance connection.

5.2 Method of Communication

Figure 5.4 illustrates our proposed algorithm for the Virtual Machine Migration. As an example, a client application (another host connected to the Internet) interacts with a server application running in VM which physically runs on VM Node 1. Identical addresses coexist in the source and destination data nodes. The two nodes are in different VLANS .The resource manager decides to migrate the VM Appliance to VM Node 2.

The steps of the migration algorithm is explained below:

- When a VM is moved, the LISP4 testbed must receive a message that contains the new destination of the VM. This message sequence can be generated in the following way. A program is installed in the VM node that captures a packet from the destination VM and will generate a message to the LISP4 testbed more specifically to the Map-Server. Specific types of packets, such as ARP or RARP (Reverse Address Resolution Protocol) packets, can be captured for this purpose.

- Just after a migration, ARP or RARP packets are usually generated. In Xen, ARP packets are generated . The program catch these packets. To do so, the program must be put somewhere in the same LAN segment that the VM uses for global communication.

- Once the ARP messages are captured, the destination VM node sends an message to its gateway of the arrival of the virtual machine. The gateway which is the ETR sends an Solicitated Map –Request to the Map-Server updating the mappings for the migrated virtual machine. The traffic is redirected to the new RLOC where the migrated machine exists .

The algorithm is given in the Fig 5.4 and the whole process of migration is explained in the Fig 5.2 , 5.3 ,5.5

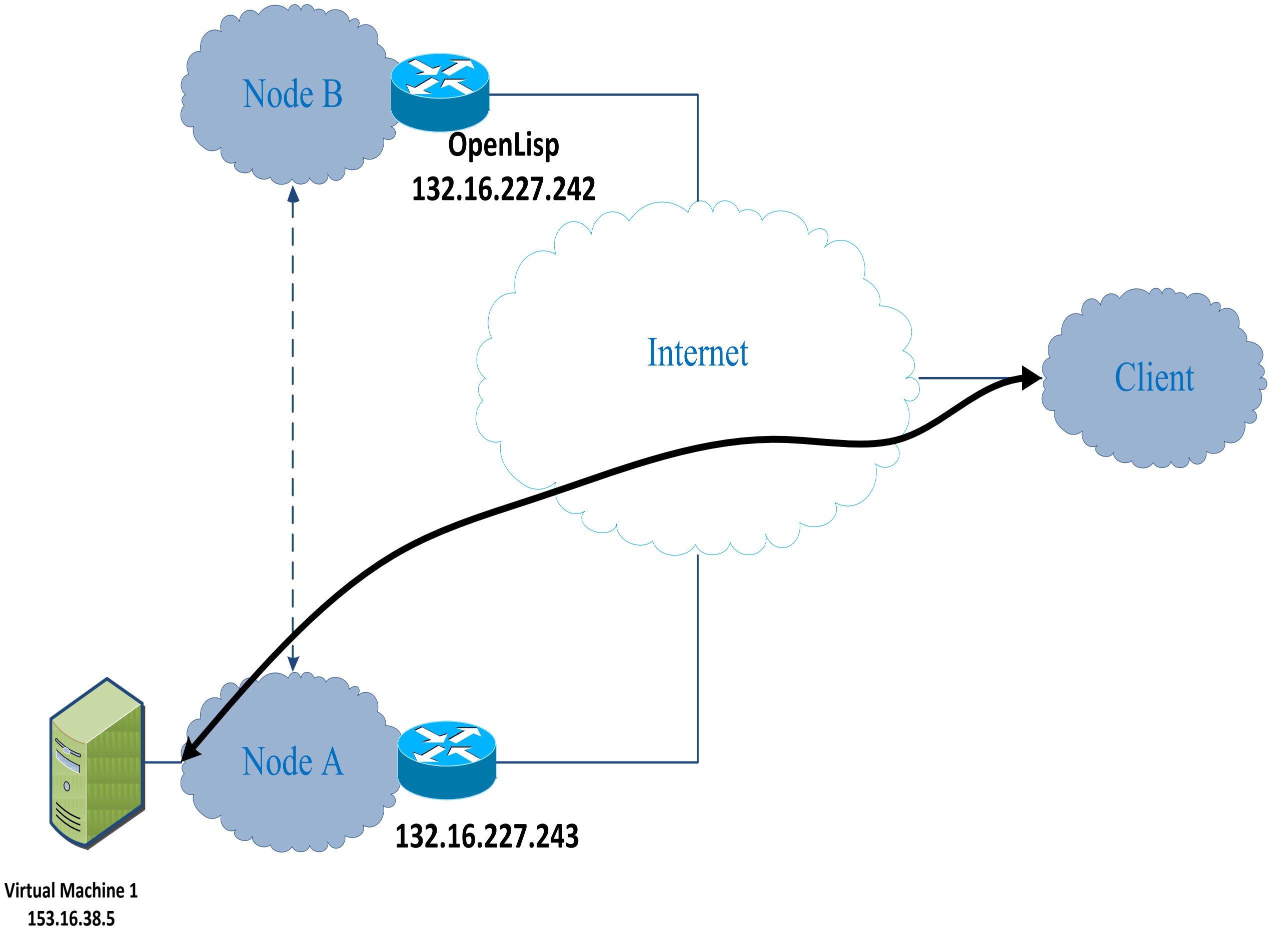


Figure 5. 2 Communication before migration

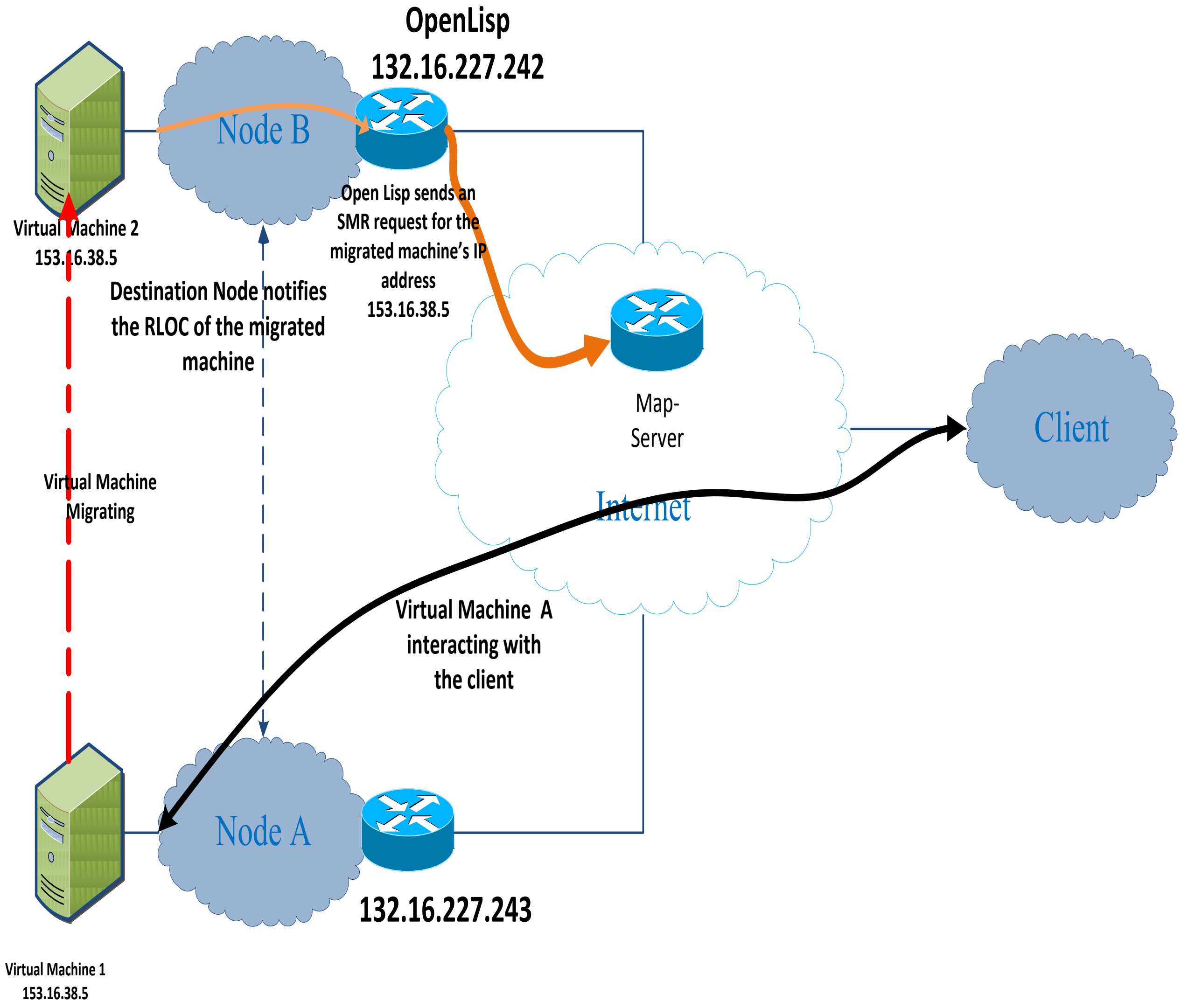

OpenLisp
132.16.227.242
Node B
Virtual Machine 2
153.16.38.5
Open Lisp sends an SMR request for the migrated machine's IP address 153.16.38.5
Destination Node notifies the RLOC of the migrated machine
Map-Server
Internet
Client
Virtual Machine Migrating
Virtual Machine A interacting with the client
Node A
132.16.227.243
Virtual Machine 1
153.16.38.5

Figure 5. 3 Communication during the migration

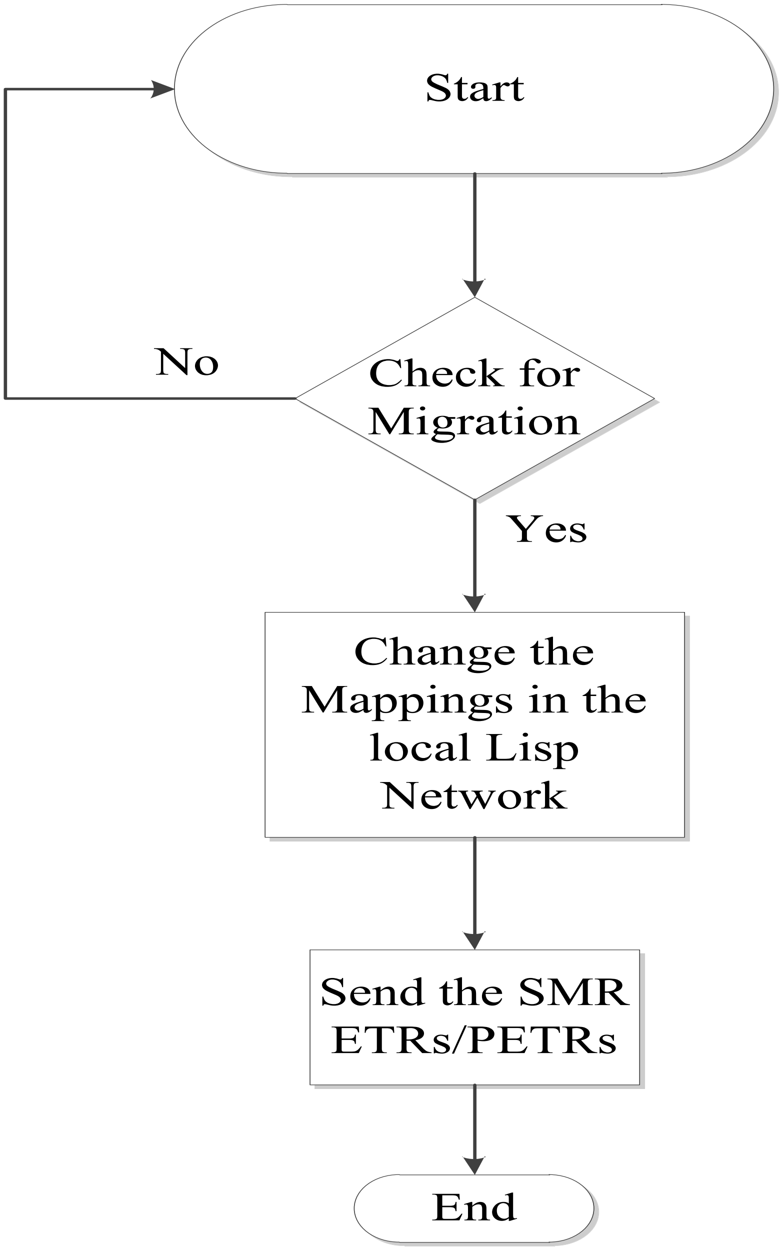


Figure 5. 4 Migration algorithm

The following procedure shows how a SMR(Solicitated Map Request) exchange occurs when a site is doing locator-set compaction for an EID-to-RLOC mapping:

1. When the database mappings in an ETR change, the ETRs at the site begin to send Map-Requests with the SMR bit set for each locator in each map-cache entry the ETR caches.

2. A remote ITR which receives the SMR message will schedule sending a Map-Request message to the source locator address of the SMR message or to the mapping database system. A newly allocated random nonce is selected and the EID-prefix used is the one copied from the SMR message.

3. The ETRs at the site with the changed mapping will reply to the Map-Request with a Map-Reply message that has a nonce from the SMR-invoked Map-Request.

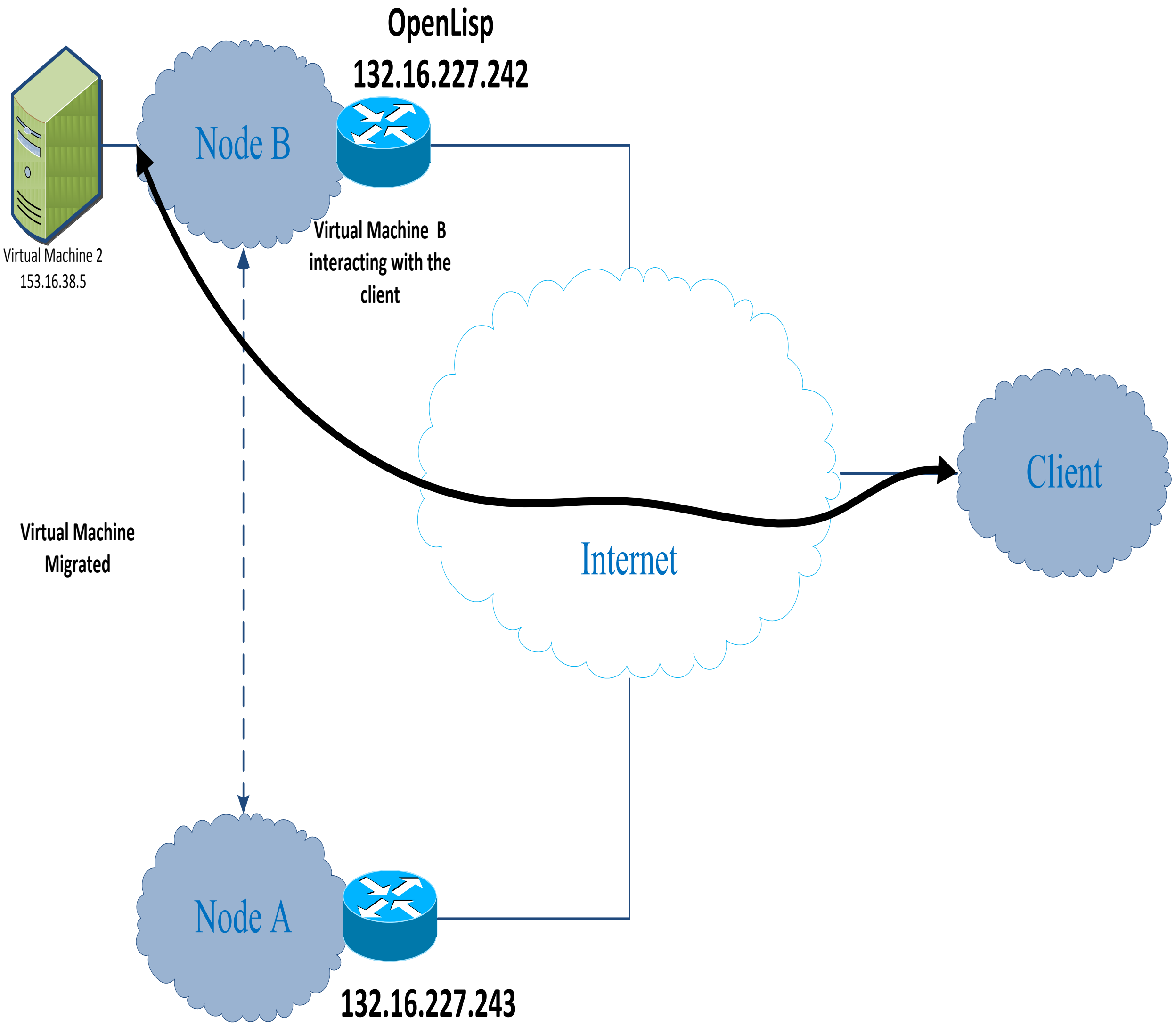


Figure 5. 5 Communication after the migration

5.3 Performance Analysis

Experiments are designed to simulate an application running on a virtual machine which is migrated over a WAN. The basic structure of the tests is to run an application on a VM, migrate the VM, and determine the amount of downtime the application experiences as a result of the migration by analyzing packets captured between the virtual machine and another host. We tested with three different applications ftp, http and video streaming that sends packet from server to a client. The method provides a convenient mechanism to assess the downtime caused by live migration.

To determine the exact downtime, we run a script in the Node 1 which would measure the migration time of the machine and we also capture the packets on the client (another host on the WAN). The application-level downtime for the VM was calculated by capturing packet dumps on the client while migration takes place. For a short period of time, the packets will be dropped. The sequence numbers embedded in the packets allows us to calculate the number of packets lost, and therefore the amount of network downtime the virtual machine experienced. The downtime here consists of the time it takes for the LISP stack to detect that it has moved to a new network, updating the map-cache of the ETR, PXTR'S to reflect the migration process and begin routing packets to the virtual machine, in addition to the small downtime always present in the final stages of Xen live migration. Since the VM is running Xen, it can be migrated as usual with “xm migrate”. The live migration of CPU and memory state is handled by Xen the same as they are on a LAN.

5.3.1 Experiment 1 – FTP

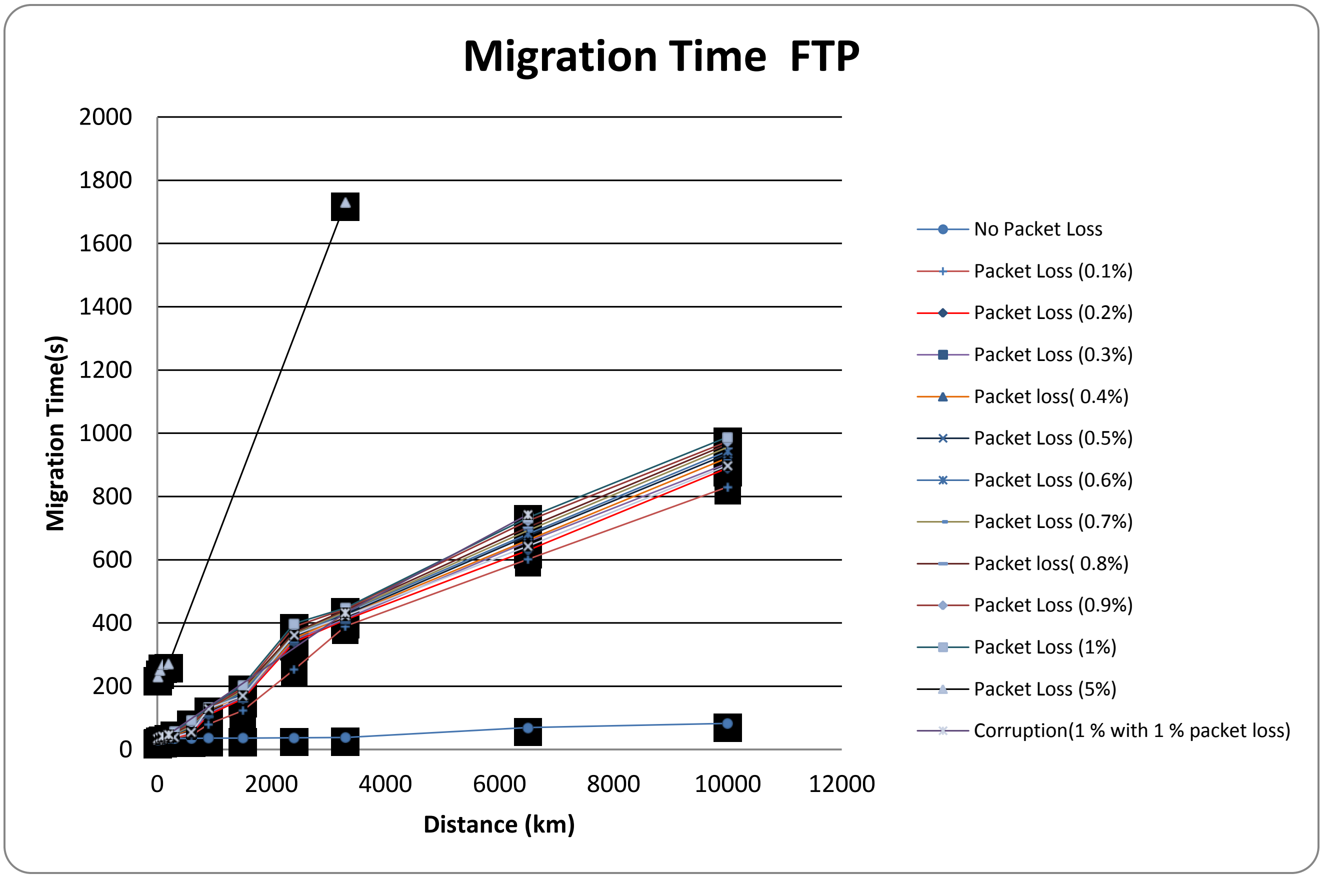


Figure 5. 6 Migration analysis of FTP for different packet loss, corruption and distance

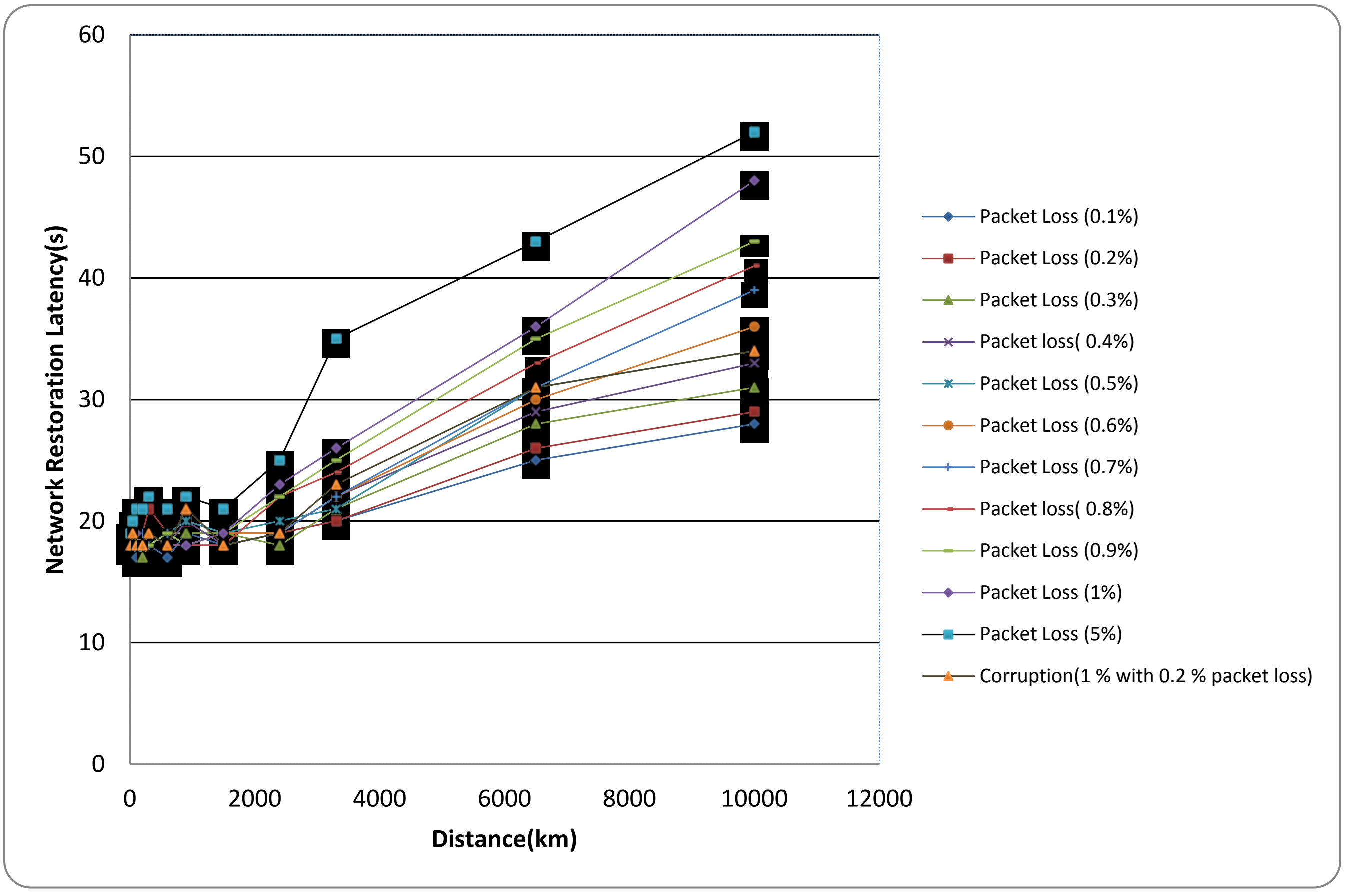


Figure 5. 7 Network Restoration Latency for FTP Application

An ftp (file transfer protocol) server is hosted in the virtual machine .A ftp file transfer is initiated from the client site. The goal of this experiment is to measure the downtime the ftp (file transfer protocol) application experiences in a stream of traffic when a virtual machine migrates to a foreign network.

The results from experiment 1 are illustrated in Figure 5.7. There is a minimum network restoration latency for 10 seconds, which is constant for the small delays , but linearly increases for the higher values of delays between the networks is increased. While this size of delay not ideal, but it is something that can be handled and recovered from by TCP, especially for less interactive applications. Fig 5.6 shows the different migration times of this application for increasing distance.

5.3.2 Experiment 2 – HTTP

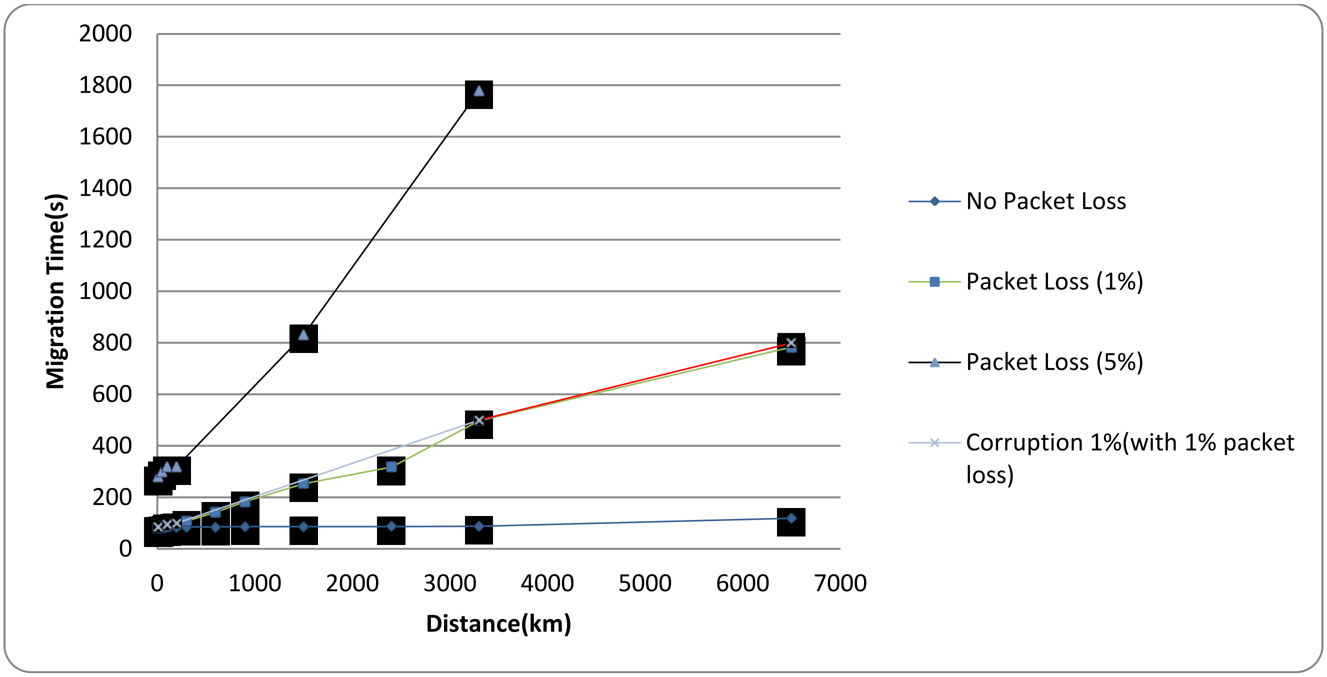


Figure 5. 8 Migration analysis of HTTP download for different packet loss (>1%) and distance

An http server is hosted in the Virtual Machine. An http file download of 500 MB is started from the Virtual Machine 1 towards a client in the Internet. The client is on a Non-LISP site. So the communication between these two endpoints happens through the Proxy ITRs (PXTRs). The results from experiment 2 are illustrated in Figure 5.8. The results clearly show that the elapsed time for the http traffic increases by a small amount, which is acceptable since the network is now being shared with an extremely busy http server workload. In spite of this increase, the download resumed once the machine migrated to Node 2.

5.3.3 Experiment 3 – Video Streaming

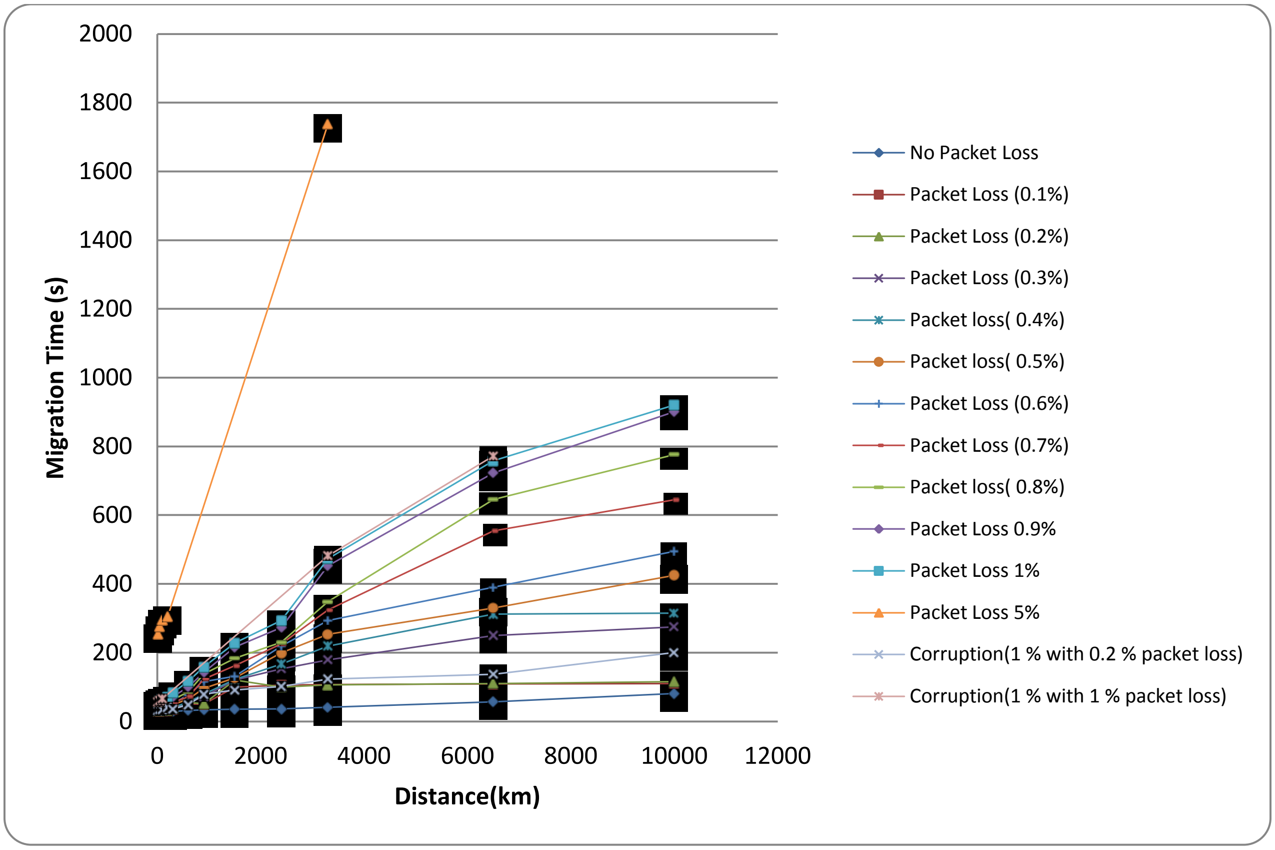


Figure 5. 9 Migration analysis of Video Streaming for different packet loss, corruption and distance

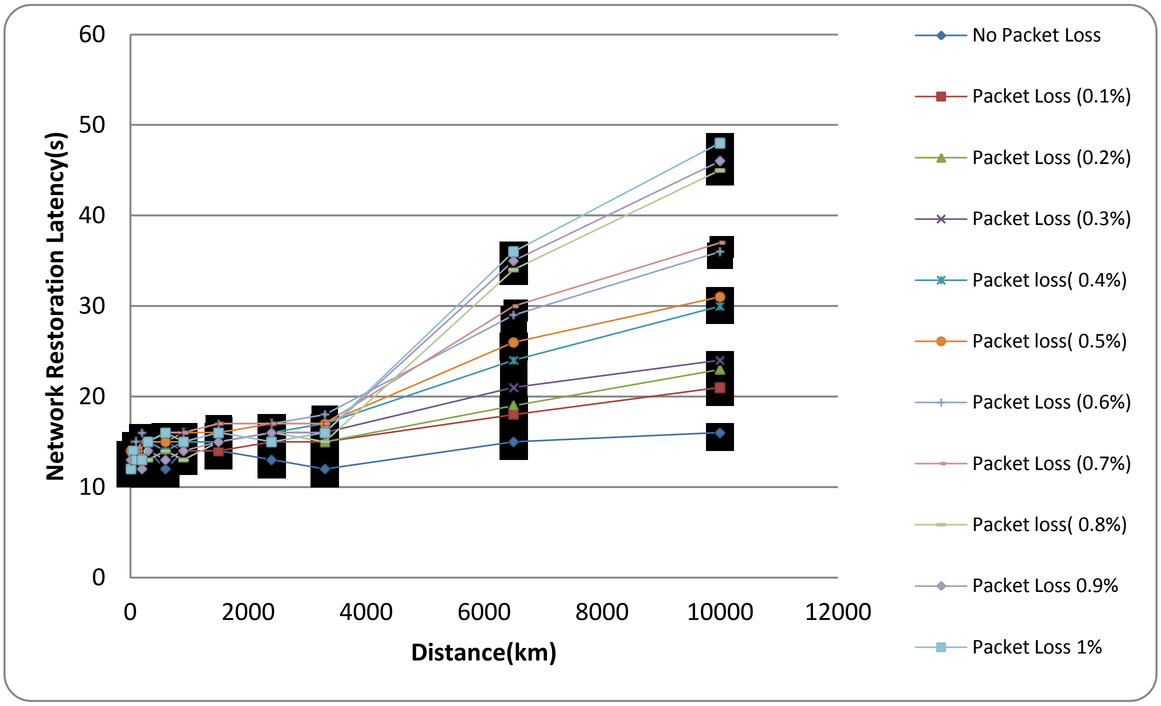


Figure 5. 10 Network Restoration latency for Video Streaming

A video streaming is started from the Virtual Machine 1 towards a client in the Internet. The client is on a Non-LISP site. So the communication between these two endpoints happens through the Proxy ITRs (PXTRs). The results from experiment 3 are illustrated in Figure 5.10. At the end of migration, no traffic is sent or received by the VM for about 10 seconds. After VM migration completes, the quality of services is restored to normal level. . Fig 5.9 shows the different migration times of this application for increasing distance.

5.4 Conclusion

In this migration technique, the problem of triangulation is not seen during the migration. The Node1 doesn’t take part in the routing for the migrated machine once the machine is migrated. This was not seen in earlier WAN migration techniques. The LISP4 testbed is updated with the new mappings for the migrated machine. Any active connections from the client are resumed after the migration. Also in this solution, the BGP routing table of the core internet is never polluted with more granular prefixes to accommodate the prefix of the migrated machine.

Chapter 6: Conclusions and Future Work

6.1 Conclusion

In this thesis, a new Internet architecture is proposed based on the consideration of efficient support of both routing scalability and mobility management. We predict the future service as a virtualization of all components so that we again apply this concept to each end-to-end communication. During the thesis, we successfully designed and implemented LISP control-plane modules for OpenLISP, an Opensource software router. Also in this thesis, we designed, implemented and validated required procedures to achieve a live virtual machine migration across distant Internet datacenters, on the Xen virtualization platform. In our Virtual Machine Mobility solution, VM migration events are dynamically detected by the destination Virtual Machine Monitor node based on data plane events. By updating the RLOC-to-EID mappings, traffic is redirected to the new locations without causing any churn in the underlying routing. The solution was validated using real-life application servers migrating across data centers while clients accessing these applications.

In our experiments, we have demonstrated that a VM can be migrated without causing loss of end-to-end connectivity. Our experiment using shared storage suggests that a migration, in the best case, takes on the order of seconds. The performance of different traffic types is evaluated assuming different distances among datacenters, as a function of the Cloud application type, bandwidth and distance between datacenters. Using LISP provides a standards-based approach for preserving the network context once a VM is migrated. A live migration over the Internet will likely have downtime orders of magnitude longer than live migration involving nodes in a single LAN. However, as long as migrations are infrequent, and as long as end-to-end connectivity can be maintained, live migration over the Internet is possible and offers potential benefits that should offset performance issues.

6.2 Future Work

Further research will be focused on transferring local persistent information more efficiently to reduce service down-time in live migration. Last but not least, secure and trusted Virtual Machine migration should be studied in the future. VM migration across distributed resources from different administrative domains brings severe security problems from resource management and monitoring perspective.

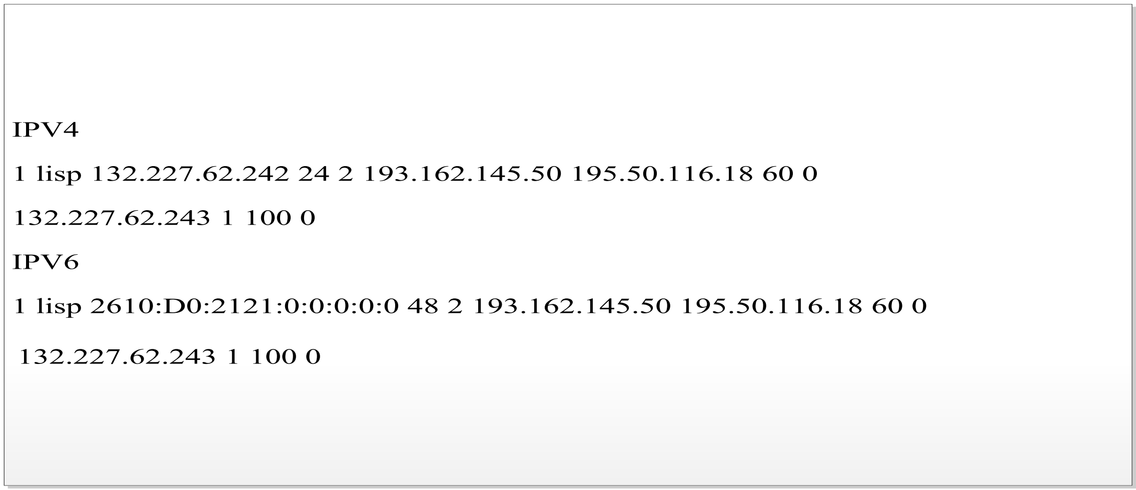
Appendix

In this section, we discuss the configurations of the tools used in the thesis. Also we present the implementation details of the OpenLISP control plane.

7.1 OpenLISP

```
IPV4

 [ RLOC count N] [key] [EID prefix] [prefix length] [Map-Server Count] [Map-Server 1] [Map-Server 2] [TTL]

 [RLOC 1] [Priority] [Weight] [Local/Self]

 .

 .

 [RLOC N] [Priority] [Weight] [Local/Self]

  IPV6

 [ RLOC count] [key] [EID prefix] [prefix length] [Map-Server Count] [Map-Server 1] [Map-Server 2] [TTL]

 [RLOC 1] [Priority] [Weight] [Local/Self]

 .

 .

  [RLOC N] [Priority] [Weight] [Local/Self]
```

Figure 7. 1 Configuration file for OpenLISP



```
IPV4
1 lisp 132.227.62.242 24 2 193.162.145.50 195.50.116.18 60 0
132.227.62.243 1 100 0
IPV6
1 lisp 2610:D0:2121:0:0:0:0:0 48 2 193.162.145.50 195.50.116.18 60 0
 132.227.62.243 1 100 0
```

Figure 7. 2 Example OpenLISP configuration file

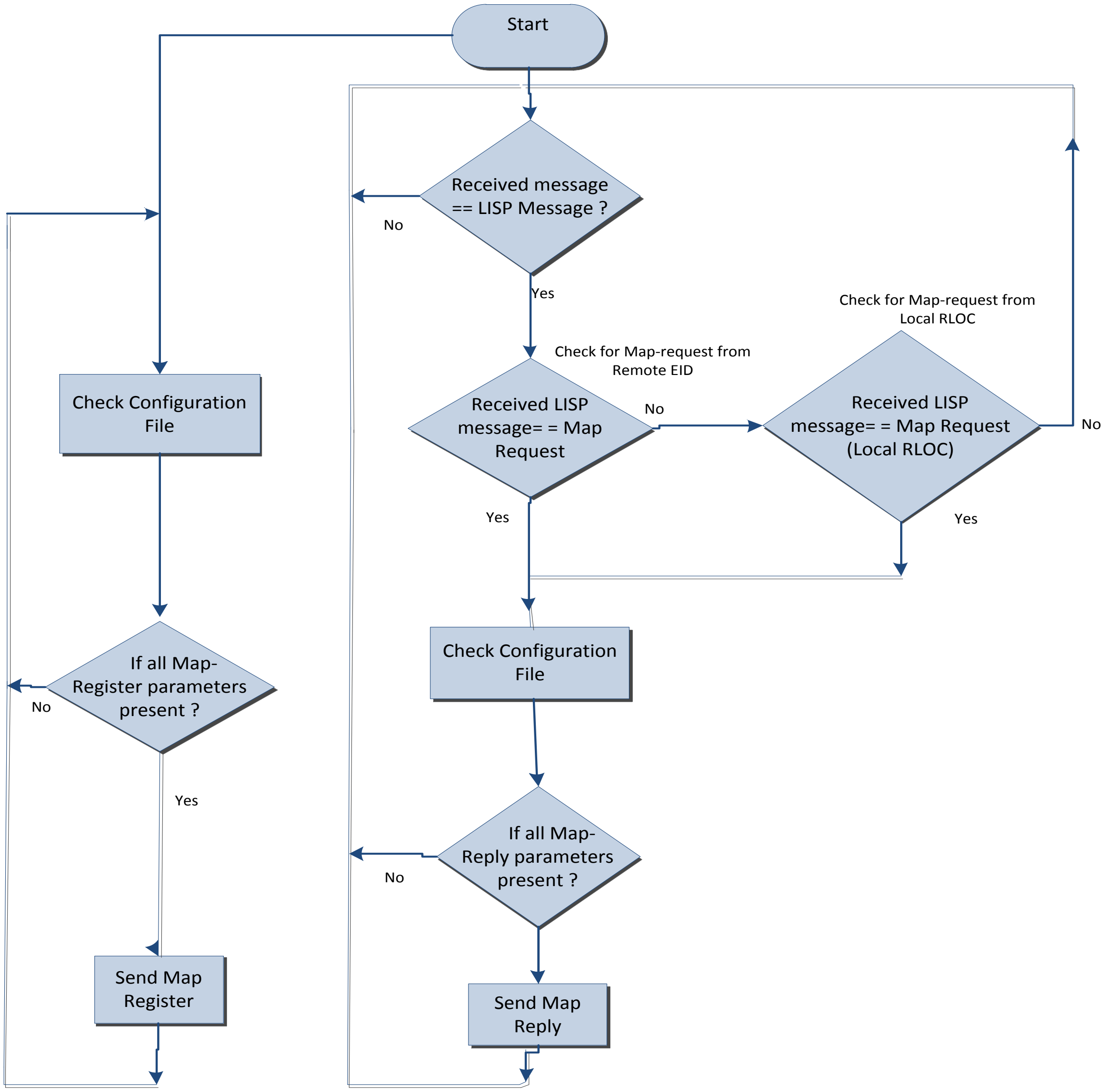
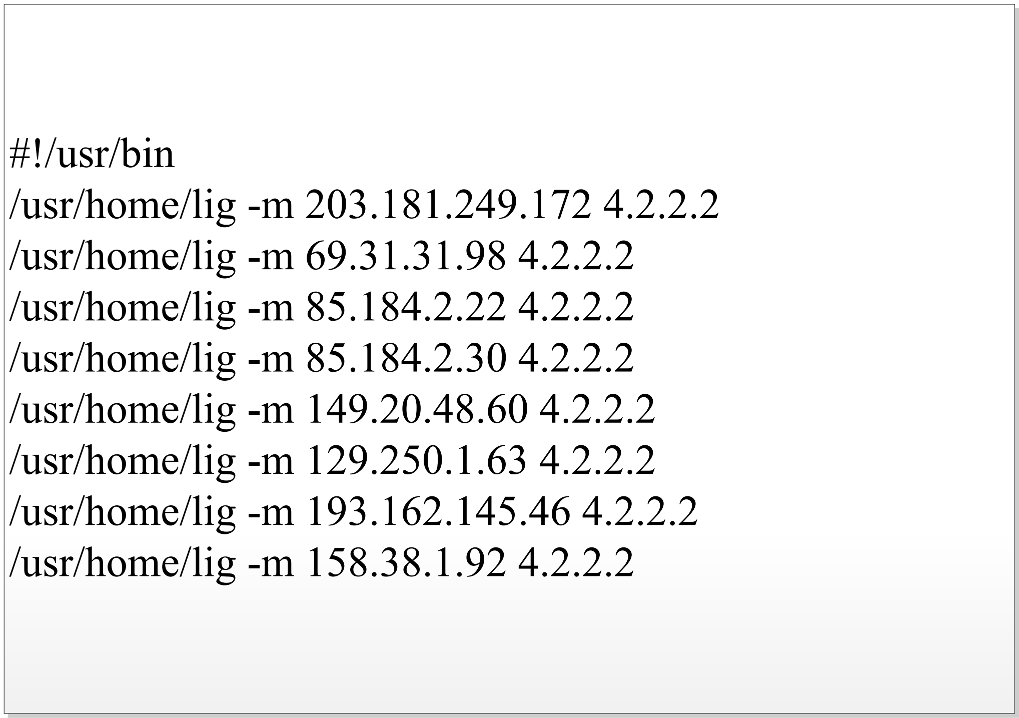

Start
Received message == LISP Message ?
No
Yes
Check for Map-request from Remote EID
Received LISP message= = Map Request
No
Check for Map-request from Local RLOC
Received LISP message= = Map Request (Local RLOC)
No
Yes
Yes
Check Configuration File
If all Map-Register parameters present ?
No
Yes
Send Map Register
Check Configuration File
If all Map-Reply parameters present ?
No
Send Map Reply

Figure 7. 3 Flow Diagram of the control plane implementation

The configuration file provides the information about the fields used in the Map-Register and Map-Reply messages. An example configuration of the configuration file is show in Fig 7.4. The implementation is done in C language.



```
#!/usr/bin
/usr/home/lig -m 203.181.249.172 4.2.2.2
/usr/home/lig -m 69.31.31.98 4.2.2.2
/usr/home/lig -m 85.184.2.22 4.2.2.2
/usr/home/lig -m 85.184.2.30 4.2.2.2
/usr/home/lig -m 149.20.48.60 4.2.2.2
/usr/home/lig -m 129.250.1.63 4.2.2.2
/usr/home/lig -m 193.162.145.46 4.2.2.2
/usr/home/lig -m 158.38.1.92 4.2.2.2
```

Figure 7. 4 Script to send the Solicitated Map Request to all the PXTR's

This script is invoked just after the migrated machine comes up in the Destination Node.

The Solicitated map-request is sent to all the RLOCs present in the OpenLISP map-cache and also to all the PXTRs.

7.2. Xen

We create a Virtual Machine with the following command:

```
xm create –c /etc/xen/VCC.cfg
```

The –c option immediately connects the console to the Virtual Machine, allowing the guest OS boot process to be monitored.

```
sudo xm migrate "$(sudo xm list  | grep WAdmin | awk -F' ' '{ print $2 }')" 10.0.0.1
```

Figure 7. 5 Xen script for migration run in the migrating Node

This script is present in the migrating node and is run to start the Virtual Machine migration. Upon completion of the migrate command, issue the xm list command. The Virtual Machine domain should not be listed. Connect to a *Domain0* command prompt on the target Xen node and issue the xm list command to verify that the virtual machine domain now resides on the target server.

Bibliography

Bibliography

[1] R. Goldberg, “Survey of Virtual Machine Research”, IEEE Computer, pp. 34-45, June 1974.

[2] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live Migration of Virtual Machines. In Proceedings of the 2nd ACM/USENIX Symposium on Networked Systems Design and Implementation (NSDI), Boston, MA, May 2005.

[3] Y. Rekhter, Ed., T. Li, Ed., S. Hares, Ed., A Border Gateway Protocol 4 (BGP-4), IETF RFC 4271, January 2006.

[4] S. Secci, K. Liu, G. K. Rao, and B. Jabbari, “Resilient traffic engineering in a Transit-Edge separated internet routing,” in ICC 2011 Communications QoS, Reliability and Modeling Symposium (ICC’11 CQRM), Kyoto, Japan, Jun. 2011.

[5] D. Farinacci, V. Fuller, D. Mayer, D. Lewis, “Locator/ID Separation Protocol (LISP),”draft-ietf-lisp-12, April. 2011.

[6] Iannone, L., Saucez, D., Bonaventure, O., “Implementing the Locator/ID Separation Protocol: Design and experience”, March 2011

[7] http://www.freebsd.org

[8] Boris Dragovic, Keir Fraser, Steve Hand, Tim Harris, Alex Ho, Ian Pratt, Andrew Warfield, Paul Barham, and Rolf Neugebauer. Xen and the Art of Virtualization. in ACM Symposium on Operating Systems Principles, October 2003.

[9] Barham, P., Dragovic, B., Fraser, K., Hand, S., and Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A., “Xen and the Art of Virtualization”, 19th ACM Symposium on Operating Systems Principles (SOSP ’03), pp. 164–177, 2003.

[10] Waldspurger, C. A., “Memory Resource Management in VMware ESX Server”, 5th Symposium on Operating Systems Design and Implementation (OSDI ’02), December 2002.

[11] Nelson, M., Lim, B.-H., and Hutchins, G., “Fast Transparent Migration for Virtual Machines”, 2005 USENIX Annual Technical Conference, pp. 25, 2005.

[12] Clark, C., Fraser, K., Hand, S., Gorm Hansen, J., Jul, E., Limpach, C., Pratt, I., and Warfield, A., "Live Migration of Virtual Machines", 2nd Symposium on Networked Systems Design and Implementation, pp. 273–286, 2005.

[13] Qin Li, Jinpeng Huai, Jianxin Li, Tianyu Wo, and Minxiong Wen, "HyperMIP: Hypervisor Controlled Mobile IP for Virtual Machine Live Migration across Networks", 11th IEEE High Assurance Systems Engineering Symposium, pp. 80–88, 2008.

[14] Haikun Liu, Hai Jin, Xiaofei Liao, Liting Hu, and Chen Yu, "Live Migration of Virtual Machine Based on Full System Trace and Replay", 18th ACM Int'l Symposium on High Performance Distributed Computing (HPDC '09), pp. 101–110, 2009.

[15] Bradford, R., Kotsovinos, E., Feldmann, A., and Schiöberg, H., "Live Wide-Area Migration of Virtual Machines Including Local Persistent State", 3rd ACM/Usenix International Conference On Virtual Execution Environments (VEE '07), pp. 169–179, 2007.

[16] Routeviews website: www.routeviews.org

[17] Y. Wang, J. Bi, J. Wu, "Empirical analysis of core-edge separation by decomposing Internet topology graph, " in *Proc. of GLOBECOM 2010*.

[18] E. Nordmark, Sun Microsystems, M. Bagnulo "Shim6: Level 3 Multihoming Shim Protocol for IPv6", IETF RFC 5533, June 2009

[19] R. Moskowitz,ICSA Labs, a division of Cybertrust, Inc.,P. Nikander,Ericsson Research Nomadic Lab "Host Identity Protocol (HIP) Architecture" , IETF RFC 4423,May 2006

[20] P. Mockapetris, "Domain names - implementation and specification," IETF, rfc 1035, Nov. 1987

[21] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[22] D. Chisnail_The Definitive Guide to the Xen Hypervisor, Prentice Hall, 2008.

[23]The Linux Foundation, Netem Network Emulator,
http://www.linuxfoundation.org/en/Net:Netem,2011

[24]Cisco IOS Command Line Interface Tutorial,
http://www.cisco.com/warpcpropub/45/tutorial.htm,1997

[25]Junos, http://www.juniper.net/us/en/products-services/nos/junos/, 2011

[26] Video Lan Organization, http://www.videolan.org/vlc/, 2011

Curriculum Vitae

Guruprasad Rao K   received the B.E. (Bachelor of Engineering) in Telecommunications in 2007 from Vishweshwariaya Technological University. He then pursued his M.S. (Master of Science) in Computer Engineering in the Department of Electrical and Computer Engineering at George Mason University. He worked as a graduate research assistant at the Communications and Networking Lab of GMU. His research interests are about future Internet routing and switching architectures.