ELECTRONIC SYSTEM LEVEL VALIDATION OF ETHERNET SWITCH

Major Project Report

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology

in

Electronics & Communication Engineering

(Embedded Systems)

By

Sunil Shah

(13MECE19)

Electronics & Communication Engineering Branch

Electrical Engineering Department

Institute of Technology

Nirma University

AHMEDABAD-382481

May 2015

ELECTRONIC SYSTEM LEVEL VALIDATION OF ETHERNET SWITCH

Major Project Report

Submitted in partial fulfillment of the requirements

For the degree of

Master of Technology

in

Electronics & Communication Engineering

(Embedded Systems)

By

Sunil Shah

(13MECE19)

Under the guidance of

External Project Guide:

Mr. Shailesh Mistry

Manager, IC Design Engineering,

Broadcom Communication Tech. Pvt. Ltd,

Banglore.

Internal Project Guide:

Prof. Ruchi Gajjar

Asst. Professor (EC Dept.),

Institute of Technology,

Nirma University, Ahmedabad.

Electronics & Communication Engineering Branch
Electrical Engineering Department
Institute of Technology
Nirma University
Ahmedabad-382 481
May 2015

Declaration

This is to certify that

a. The thesis comprises my original work towards the degree of Master of Technology in Embedded Systems at Nirma University and has not been submitted elsewhere for a degree.

b. Due acknowledgment has been made in the text to all other material used.

- Sunil Shah

Disclaimer

The content of this report does not represents the technology, opinions,beliefs or positions of Broadcom Communication Technologies Pvt. Ltd., its employees, vendors, customers or associates.

Certificate

This is to certify that the Major Project entitled **"ELECTRONIC SYSTEM LEVEL VALIDATION OF ETHERNET SWITCH"** submitted by **Sunil Shah (13MECE19)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in Embedded Systems, Nirma University, Ahmedabad is the record of work carried out by him under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination.The results embodied in this major project, to the best of our knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Date: Place: Ahmedabad

Prof. Ruchi Gajjar
Internal Guide

Dr.N.P.Gajjar
Program Coordinator

Dr.D.K.Kothari
Section Head,EC

Dr.P.N.Tekwani
Head of EE Dept.

Dr.K.Kotecha
Director, IT

-

Certificate

This is to certify that the Major Projec **"ELECTRONIC SYSTEM LEVEL VALIDATION OF ETHERNET SWITCH"** submitted by **Sunil Shah (13MECE19)**, towards the partial fulfillment of the requirements for the degree of Master of Technology in Embedded Systems, Nirma University, Ahmedabad is the record of work carried out by him under our supervision and guidance. In our opinion, the submitted work has reached a level required for being accepted for examination.

Date: Place: Bangalore

Mr. Shailesh Mistry,
Manager,IC Design Engineering,
Broadcom Communication Tech. Pvt.Ltd.
Bangalore.

Acknowledgements

I would like to express my gratitude and sincere thanks to **Dr. K. Kotecha**, Director, Institute of Technology,Nirma University, **Dr. P. N. Tekwani**, Head of Electrical Engineering Department, **Dr. Dilip Kothari**, Section Head,EC and **Dr. N. P. Gajjar**, Coordinator of M.Tech Embedded Systems program for allowing me to undertake this thesis work and for their guidelines during the review process.

I am deeply indebted to my thesis supervisors, **Mr. Shailesh Mistry**, Manager, Broadcom Communication Technologies Pvt. Ltd. and **Prof. Ruchi Gajjar**, Assistant Professor, EC Department, Nirma University for their constant guidance and motivation. I also wish to thank all other team members at Broadcom for their constant help and support. Without their experience and insights, it would have been very difficult to do quality work.

I wish to thank my friends of my class for their delightful company which kept me in good humor throughout the project.

Last, but not the least, no words are enough to acknowledge constant support and sacrifices of my family members because of whom I am able to carry out the project work successfully.

- Sunil Shah

13MECE19

Abstract

With the increasing level of complexity in digital systems, digital design techniques have advanced and got to a higher level of abstraction that is Electronic System Level (ESL). Designing at the ESL tackles the increasing complexity of System on Chip (SoC) design by raising the level of abstraction in system specification and modelling which also helps to decrease the time to market of the chip. This process involves validating and transforming of the system model with the effective ESL frameworks until the desired level of SoC implementation and coverage is attained. These optimized switches are used, to meet scalability, bandwidth and efficiency demands of the networking environment. This project aims at validating various features at data link layer, network layer and overlay networks of Ethernet switch. These features includes virtualization protocols like VXLAN, NVGRE, etc. Data center applications are no longer bound to specific hardware resources; thus making the application unaware of the underlying hardware and viewing the CPUs, memory, and network infrastructure as shared resource pools. Server virtualization allows the abstraction of server resources to provide flexibility and optimized usage on a standardized hardware infrastructure. Results are analysed for a match with the expected behaviour of the switch. Bugs found in the software model are reported back to the implementation team for a fix and a bug free software model of the switch is obtained for Register Transfer Level (RTL) implementation.

Contents

List of Tables

List of Figures

Abbreviation Notation and Nomenclature

ESL ..... Electronic System Level
SoC ..... System on Chip
RTL ..... Register Transfer Level
HLS ..... High Level Synthesis
VXLAN ..... Virtual eXtensible LAN
EDA ..... Electronics Design Automation
NVGRE ..... Network Virtualization using Generic Routing Encapsulation
TCL ..... Tool Command Language
CAM ..... Content Addressable Memory
VLAN ..... Virtual Local Area Network
IP ..... Internet Protocol
STP ..... Spanning Tree Protocol
BPDU ..... Bridge Protocol Data Unit
MSTP ..... Multiple Spanning Tree Protocol
ARP ..... Address Resolution Protocol
RU ..... Rack Unit
VM ..... Virtual Machine
NVE ..... Network Virtualization Edge
NVA ..... Network Virtualization Authority
VNI ..... Virtual Network Identifier
VTEP ..... Virtual Tunnel End Point
VSID ..... Virtual Subnet Identifier
GRE ..... Generic Routing Encapsulation
PA ..... Provider's Address

Chapter 1

Introduction

1.1 Motivation

As chip design complexity continues to increase, validation methodologies attempt to keep the highly complicated task of validation manageable. Instead of delaying the process of detecting the bugs in Register transfer level (RTL) code, the design flow is accelerated to achieve bug-free RTL designs. This is achieved by automating the generation of RTL from exhaustively verified system software models using ESL validation. High-level synthesis (HLS) can then produce RTL that matches the high-level source specification and is free of the errors introduced by manual coding[4]. RTL verification task will be made much easier, and the time to verified RTL will be significantly reduced. It becomes very much easy to validate at software level as compared to hardware level such as RTL and time taken to validate is also very less as it is at more abstract level as compared to hardware.

An Ethernet switch in computer networking is connected to multiple devices and servers to a Local Area Network(LAN). The broadband internet connection is shared between multiple devices in the LAN. A Gigabit ethernet switch functions in the same manner, with the increased speed as compared to standard ethernet. Gigabit Ethernet transmits at approximately one gigabit per second, which is 10 times

faster than the Fast Ethernet, which transfers data at approximately 100 megabits per second. The gigabit switch is designed to work at these faster speeds, without signal loss or transfer rate reduction. Currently it also supports the speed of 10Gbps,25Gbps and 40Gbps.
There has also been increasing demand of server virtualization to utilize the server at its maximum capacity which helps in running the applications independently. The data to be stored has also moved into cloud.

1.2 Problem Definition

This project is carried out to achieve following objectives:-

a. Study the layer 2/layer 3, tunneling features supported by the Ethernet Switch, understand the methodology and develop test cases to validate the behavior of the software model of the Ethernet Switch.

b. Study the architecture of switch used in Data Center network and Overlay Network technologies such as VXLAN, NVGRE.

c. Report back the bugs found in the software model during validation to the implementation team for a fix.

1.3 Thesis Organization

Chapter 2 gives the Literature Survey of the project.
Chapter 3 describes the Validation Environment used in the project.
Chapter 4 describes gives us the Overview of Data Center Network[11].
Chapter 5 describes about VXLAN, NVGRE - Layer 2 Over Layer 3 Protocols[9, 10].
Chapter 6 describes Test scenarios validated during the project.
Chapter 7 gives the Conclusion and Future Scope of the work carried out.

Chapter 2

Literature Survey

2.1 Electronic System Level methodology

The General system design process for ESL synthesis methodologies follows a top-down approach. The hardware, software and the required synthesis steps are all designed simultaneously which is depicted by the double roof model shown in Figure2.1 below.

The ideal top-down design for embedded hardware/software systems is shown in this double roof model. The Software design process is shown on the left side of the model, whereas the hardware design process is shown on the right side model.

Each side is organized in different abstraction levels, e.g., task and instruction levels or component and logic levels for the software or hardware design processes, respectively. There is one common level of abstraction, the ESL, at which we cannot distinguish between hardware and software. At each level, in a synthesis step (vertical arrow), a specification is transformed into an implementation. Horizontal arrows indicate the step of passing models of individual elements in the implementation directly to the next lower level of abstraction as specifications at its input.

Over the past three decades, the entry point for hardware design has moved upward in the abstraction hierarchy from hand-drawn schematics to gate-level design, to

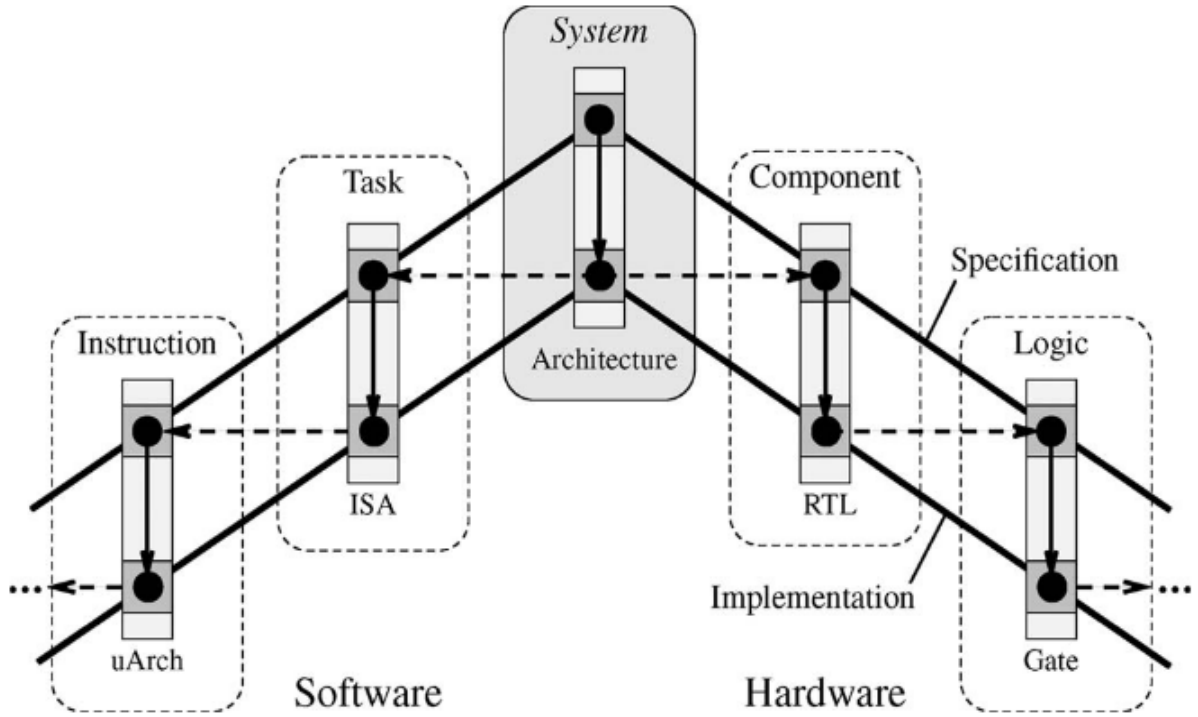

Figure 2.1: Double Roof Model[1]

-

RTL descriptions. As the hardware design complexity becomes increasingly unmanageable, finding ways to design at higher abstraction levels, and developing tools to automatically create the circuit layouts have gained more importance in industry and academia[2].

The complex SoC systems that manages large amount of data cannot be analysed on its architecture only through the RTL flow. The time taken to verify and analyze the full chip is much more; a very slow process and much expensive too. The amount of resources used to verify using RTL are much more and complexity of system has increased to such an extent that RTL cannot independently address the issues. So to carry out validation at a increased speed the model needs to be abstracted much before in the development stage.

ESL has found its way into the mainstream EDA industry in the past few years be-

cause of the increased interest in finding a new abstraction level for the design entry point of the electronic systems design process. ESL design employs methodologies and tools that help to tackle the growing complexity of designing modern electronic products. Companies can achieve significant time-to-market benefits by incorporating ESL design into their product development flows. These benefits include reduction of the product design cycle time, increased hardware design productivity, faster creation of derivative products, improved communication between hardware and software teams, and improved product quality. By using ESL design, companies can deliver better, more complex electronic products, and get them to market sooner, with a lower risk. Product development flows that incorporate ESL design are rapidly becoming a necessity in today's competitive market.

Earlier the hardware and software were designed independently withe the use of EDA and software development tools which was a sequential manual process. At each and every stage of product development the methodologies and tools that support ESL design have helped us to achieve the architectural differences significantly. These methodologies and tools are used to optimize and evaluate the product extensively. These tools and methodologies have helped to corelate the ESL model and the hardware implementations.This ensures a well-connected ESL-to-implementation design flow. ESL is all about tools and methodologies that offer the next level of productivity required for SoC design[4].

ESL validation is the task of validating ESL designs at a high abstraction level. ESL methodology is being used for development and testing of a High Speed Ethernet switch that supports Layer-2 switching, Layer-3 routing, and Tunneling services. The high-level synthesis(HLS) have helped us to create the optimize hardware from the specification using the ESL design method. With the help of HLS, the RTL hardware is automatically generated from the software model. Since the generation of RTL flow is automated, the verification effort is greatly reduced[3, 4].

Strong market trends and new workloads are driving fast, flat, and fat network designs and the server interfaces needs high speed ethernet. The popularity of dense

10GbE and 40GbE connectivity to aggregation and access layer is due to virtualization which tends to increase the utilization in the data-center network.
A high-performance and general-purpose interconnect technology is required to realize flexible and highly reliable systems consisting of network-connected fast servers with large storage capacity. High Speed Ethernet is considered promising as a standard solution to meet these requirements. These next-generation switches have been designed to address performance, capacity, and service requirements for data centers, cloud computing applications, enterprise campus backbone equipment, and mobile core networks.

2.2 Layer-2 Switching

The incoming frames are parsed and analyzed by the switches, which makes the forwarding decisions based on information contained in the frames, and forward the frames towards the destination. In this layer, switches forward and flood traffic based on Medium Access Control (MAC) addresses. In Layer-2, switch performs the following functions:

a. MAC address learning:
   The switch learns a MAC address and stores it in Content Addressable Memory (CAM) Table with corresponding port number and VLAN ID. When any switch port receives any frame, before forwarding the frame to the destination, it retrieves the source MAC address from the frame and updates the CAM Table. MAC Addresses are stored in a CAM Table for a particular time. If communication with a MAC address does not occur up to that particular time, ageing occurs and that MAC address is removed from the CAM Table depending on the counter value set for ageing process.

b. Frame forwarding:
   The switch forwards a frame based on its MAC, the switch learns the Source

MAC address with its VLAN ID and port number. Then it searches the CAM Table for the destination MAC address. There are several options for destination MAC address in CAM Table, as follows:

(1) Destination MAC address is unicast and available in the CAM Table (known unicast): If the Destination MAC is present in the CAM Table, then the switch forwards the frame to the corresponding port number and VLAN ID.

(2) Destination MAC address is unicast and not available in the CAM Table. (unknown unicast): If the destination MAC is not present in the CAM Table, the switch forwards the frame to all active ports except the receiving port, which is known as flooding. The end device decides to accept or reject the particular frame.

(3) Destination MAC address is a multicast or broadcast address: If the switch finds a multicast frame on its receiving port, it forwards the frame to all the multicast ports. The switch identifies the frame as a multicast frame by checking the 40th bit of its destination MAC address: If bit is set to 1, it is a multicast frame, otherwise it is a unicast frame. The switch also checks for broadcast frames and forwards them to all active ports except the receiving port. For broadcast frames, all bits in the destination MAC address are 1, that is FF:FF:FF:FF:FF:FF.

2.3 Virtual Local Area Network(VLAN)

VLAN is the technology that allows us to separate the users into the small individual segments even though these segments are connected to the same physical network. The hosts in two different segment may be still be connected to same physical network but if they want to communicate, they have to route the frame. So, At the MAC layer the members connected to same VLAN can only communicate within

themselves and not with other members of different VLANs. So it seems that LAN is virtual because although the stations seems to be connected to single physical network but cannot send frames between each other.
The significant advantages acheived by implementing VLANs on the network:-

a. VLANs helps to control traffic:-
   It has basically restricted the broadcast domain. Some members doesn't require some information but we unnecessarily flood it and create congestion in the network.

b. VLANs ease the change and movement of devices:-
   Since it is software configurable it has become very easy process to change the postion of the devices.

c. VLANs provide security:-
   Since two members from different VLANs cannot communicate so we can group the members accordingly which needs communication between the other devices and create separate VLAN accordingly.

2.3.1 Different Types of VLAN Assignment

a. Port Based VLAN Mapping

b. MAC Based VLAN Mapping

c. IP Subnet Based VLAN Mapping

d. Protocol Based VLAN Mapping

2.4 Spanning Tree Protocol

Spanning Tree Protocol is a network protocol is implemented to ensure a loop-free topology for the network. Loops in the network are formed due to redundant links

towards a station present in the network. These redundant links are much useful at the time of failure of the primary links so that user doesn't get interrupted during communication. So we should not remove the redundant links in the network.

The links connected in the network have weights assigned to them. Whenever

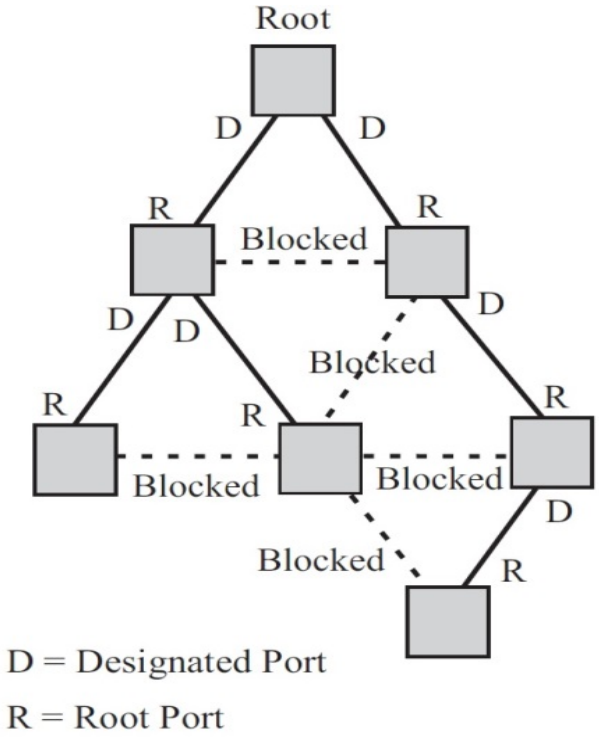

Figure 2.2: Spanning Tree Protocol

there is any change in the topology of the network, switches exchange control frame known as Bridge Protocol Data Unit(BPDUs) to intialise the network. One of the switch in the topology is elected as root switch which may be the logical center of the topology which is elected having the highest weight among all the links. If there is tie between two switches for the root bridge then the tie is broken with the switch having lower MAC address. All the other switch now calculates the shortest path towards the root bridge. The port which takes them towards the root bridge is called the root port and the port which takes away from root bridge is called the

designated port. All the other ports which are not the root port or designated port are blocked.

Figure 2.2 shows the links being blocked after the application of STP.

With the improving technology, STP has been improved in certain specific areas, One of them is Multiple Spanning Tree Protocl(MSTP). We are having certain redundant links. These redundant links can be utilised by station of some other VLAN and so the all the network links gets utilised, keeping the backup link too. So utilisation of the network is increased by using the links that are blocked by some VLANs can be used by other VLANs.

2.5 Layer-3 Switching

Since Layer-2 device needs to learn the addresses of each and every device which is appraoching few billion which is not practically feasible. Also Layer-2 needs to configure with STP and if all the device are connected at Layer-2 domain it would become impossible to track the block links and ports. So there is need Layer-3 switching device which is not local minded.

IP protocol is used to transmit the frame from source to destination where source

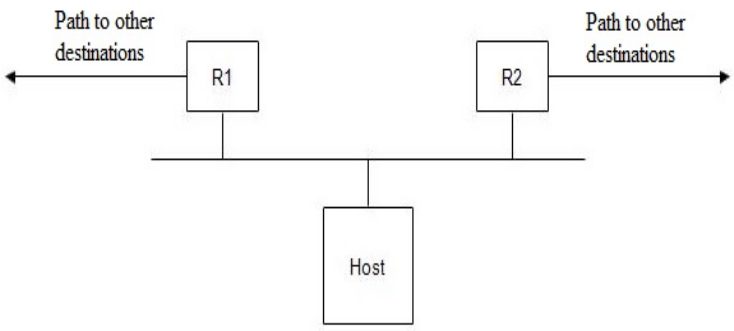

Figure 2.3: Example of host which has to make choice between the two Routers

aand destination are identified by fixed length of addresses. During this transmission, the frame has to pass through many physical networks and it seems that packet has been transmitted from source to destination without any direct connection. There are many routers in the network. For a given router there can be two or more direct connection to any other routers as shown in Figure2.3. The router has to choose to which it should forward. ARP protocol is used to find the physical address of the device in the network. IP Forwarding table helps the router to forward the packet in a particular direction. This forwarding table maintains the address of each and every possible destination on the network. But the table size is not large enough to store the address of each and every destination. So there is a addressing scheme which would help this table to store the address of all the destinations. This addressing scheme allocates the address to a small segment of the network with some common prefix. This address prefix are stored in IP forwarding table which gives us subsequent next hop address. In this way the frame is transmitted from one place to another.

The concept of forwarding tables is explained using the Figure 2.4. This example contains four networks and three routers. The forwarding table formed is for the router R taken as a reference.

As seen from the figure that router R is directly connected to the networks 20.0.0.0

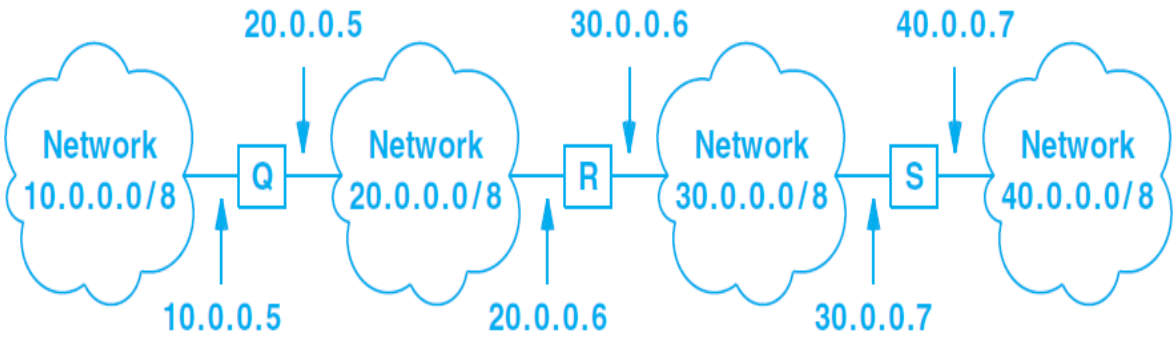

Figure 2.4: Example Internet with 4 Networks and 3 Routers

Table 2.1: Forwarding table for Router R

Destination IP Address	Forward to this IP Address
20.0.0.08	Deliver Directly
30.0.0.08	Deliver Directly
10.0.0.08	20.0.0.5
40.0.0.08	30.0.0.7

and 30.0.0.0, the frame can be directly delivered to these networks. Now suppose that router R wants to send the frame to host on the network 40.0.0.0 but since it cannot be delivered directly it forwards the frame to router S to the address 30.0.0.7 and now as router S is directly connected to network 40.0.0.0 it can send the frame directly. The number of networks in the internet will decide the size of forwarding table as seen from the above example. The size of the table is independent of the number of host connected to the network and it only increases as the new networks are added. So following things are required to transmit the frame for each entry in the forwarding table:

a. IP address.

b. Prefix mask to idnetify the network.

c. IP address of the next-hop router.

d. Network interface for transmitting.

2.6 IP Tunneling

Tunnel is a mechanism used to ship the other protocol across the network that normally wouldn't support it. So, the IP tunnel is mechansism to send the IP protocol over the other IP protocol infrastructure. IP Tunneling mechanism is used to connect two distant common IP networks which doesn't have routing path via another IP network. This mechanism in turns provide security by hiding the addresses of

sender and receiver.

The mechanism is to add an IP header on top of the existing IP header. The outer IP header source and destination identify the endpoints of the tunnel. The inner IP header source and destination identify the original sender and receiver of the frame. So if the backbone network is of IPv4 then frame is encapsulated with the IPv4 header to carry over IPv4 infrastructure or if the backbone network is of IPv6 then frame is encapsulated with the IPv6 header to carry over IPv6 infrastructure.

The significant advantage achieved by use of the IP tunnels is several hosts have migrated to IPv6 protocol although the infrastructure still exists in the IPv4 network. This has facilitated the smooth transition from IPv4 to IPv6 protocol.

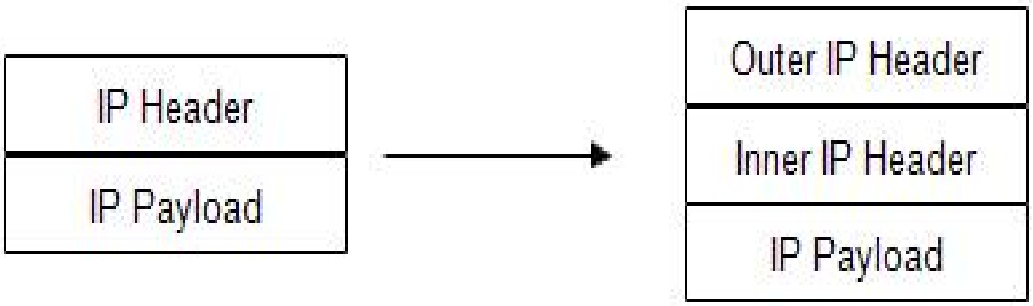

Figure 2.5: IP Tunnel

2.6.1 Types of IP Tunnels

a. 4in6 IP Tunneling

b. 6in4 IP Tunneling

c. 4in4 IP Tunneling

d. 6in6 IP Tunneling

Chapter 3

Validation Environment

3.1 Design Flow

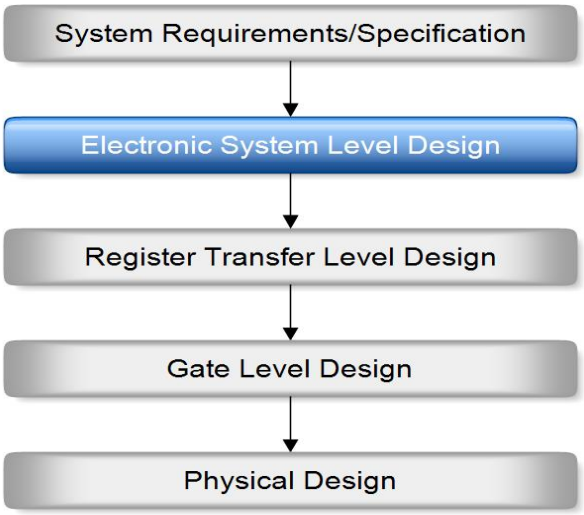

Figure 3.1: Design Flow

-

Typical IC design flow in shown in Figure3.1. Electronic system level design must be checked whether it meets the functional specification and the purpose. This is done by Electronic system level validation. Validation is the exercising of the design

to check that it is fit for purpose. It is a subjective process of using the design, to see if it does what you need. The specification is not golden and in effect is under test along with the design. The aim is to prove that the design and the specification meet purpose.

Figure3.2 shows the block diagram of ESL validation model used in our environment.

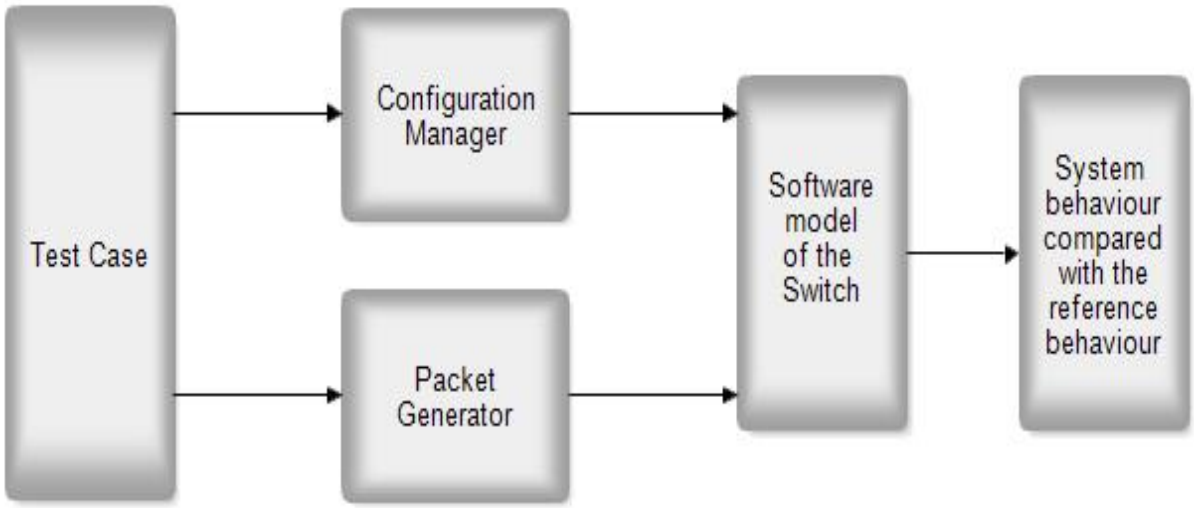

Figure 3.2: Block Diagram of ESL Validation Model

-

a. Test Case:-

Test cases for validating the software model of the chip are developed using scripting languages like Tool Command Language (TCL) and these cases are executed in the UNIX platform. Test cases contain transmitting packets, expected packets and the configurations which have to be dumped into the software model of the chip.

b. Packet Generator:-

The packet generator picks up the parameters from the TCL tests and generates or processes the packets. The packet generator is interfaced with TCL for

the test generation purpose. This TCL interface can supply input as well as it could monitor the response from the software model. The packet generator is used to generate the traffic for the validation environment.

c. Configuration Manager:-

The configuration manager provides the configuration information to the software model of the chip. The test cases and the configuration settings are given to the software model or System C model of the chip and the received packets from the model are compared with the expected packets to validate the functionality of the model.

d. Software Model of the Switch:-

Architecture specification is executed as a software model. Software model is written in SystemC. SystemC is a C++ class library and a methodology that you can use to effectively create a cycle-accurate model of software algorithms, hardware architecture, and interfaces of your SoC and system-level designs.The test cases and the configuration settings are given to the software model or System C model of the chip and the received packets from the model are compared with the expected packets to validate the functionality of the model.

3.2 Summary

This chapter describes the Validation Environment of the chip design flow.

Chapter 4

Overview of Data Center Network

4.1 Data Center Architecture

The data center is home to the computational power, storage, and applications necessary to support an enterprise business. The key things to be kept in mind while designing the data center are flexibility of quickly deploying the new services, scalability, performance and resiliency.
For designing such a data center network to achieve all the benefits mentioned above the key areas to look into are density of the port, server capacity, access layer uplink bandwidth. Data center network consists of core, aggregation and access layers. These layers are briefly described as follows:

a. Core layer:-

   The high speed packet switching is done at this layer. Several aggregation modules are connected to this layer only and it has strong Layer 3 routed fabric. Several routing protocol are running in this layer like OSPF/EIGRP. It also segregates traffic between core and several other aggregation layers.

b. Aggregation layer:-

   The several important functions like spanning tree processing, Layer 2 domain

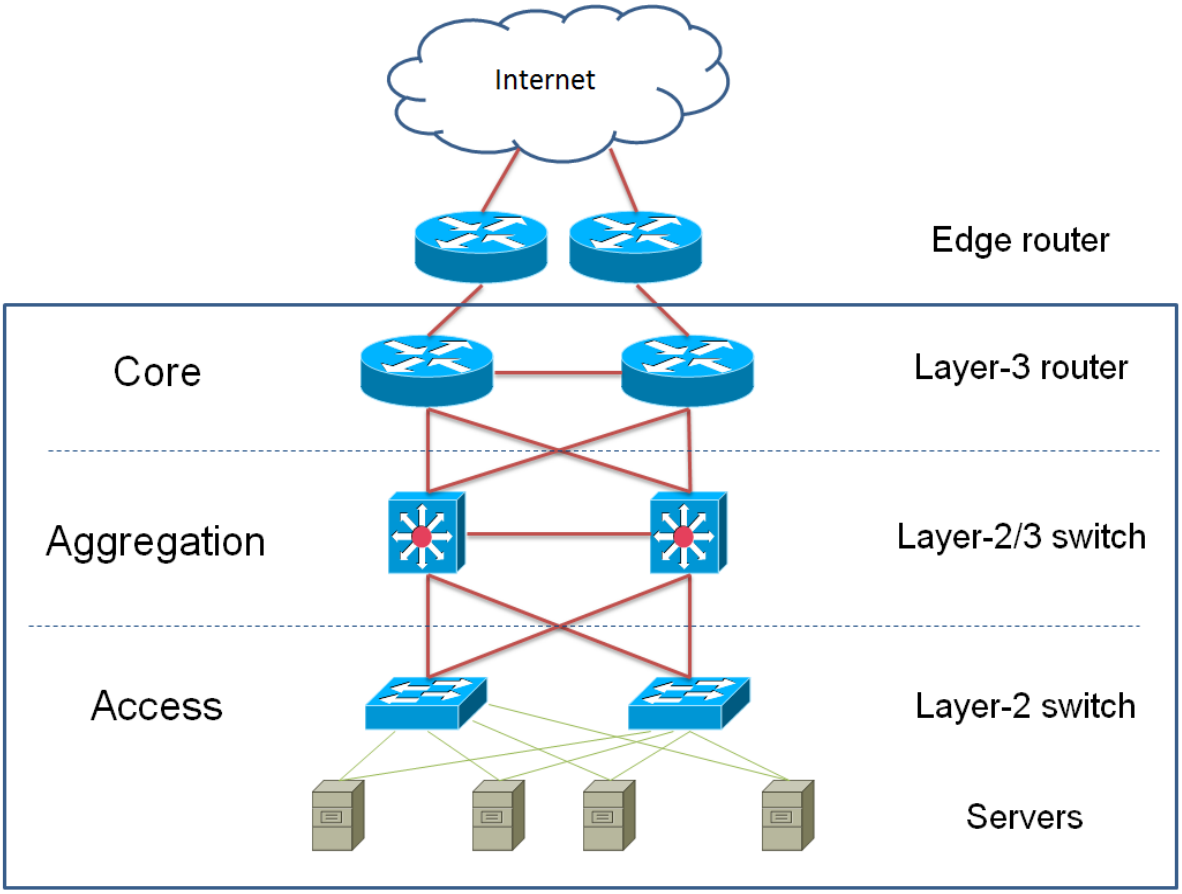

Figure 4.1: Data Center Architecture

definitions, default gateway redundancy and module integration are done at this layer. If some server has more traffic then it can offload traffic to other server through this layer.

c. Access layer:-

This layer provides both the Layer 2 and Layer 3 topologies. Servers are physically attached to the network through this layer. Typical examples are ToR switch or EoR switch. Switches provide both Layer 2 and Layer 3 topologies, fulfilling the various server broadcast domain or administrative requirements.

To increase the computation efficiency, store huge amount of data and provide all the network resources for the applications the physical resources are abstracted from their logical representation is what is known as server,storage and network virtualization respectively. There is huge demand for virtualizing the server in the data centers.This is because of the numerous benefits such as single physical server supports large VMs, more utilization, less power usage, security, reduce user downtime. Each tenant can create multiple virtual network instances. A single tenant run multiple virtual instances individually. So there is one-to-many mapping between the virtual instances and the tenants. since the virtual instances are running the application simultaneously they should be isolated from other virtual instances with the traffic to be crossed between each other only allowed by some policy.
The two issues that must be looked into specifically for data center virtualization are the following:-

a. There should be support for address space separation among the tenants.

b. The VMs in the data center should be allowed to move freely anywhere, without imposing any restriction to match the subnet boundaries of the data center.

4.2 Requirements of Modern Data Center

a. Mobility of Virtual Devices

b. Scaling of Forwarding Tables

c. Scaling of Network Segments

d. Coupling of Physical and Logical Connectivity

e. Coupling of Infrastructure and Policy

f. Virtualized Networks

g. Optimal Forwarding

h. Reduction in Dependency on Traditional Protocols

4.3 Overlay Networks

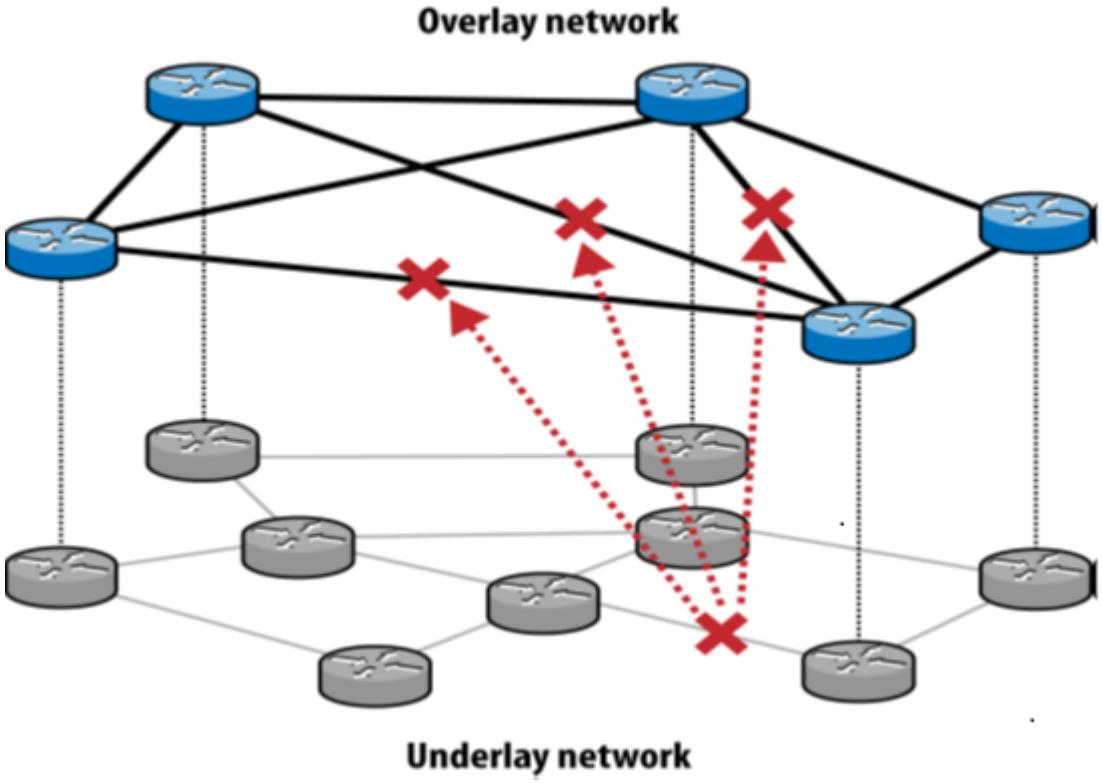

Figure 4.2: OverlayNetwork

Figure4.2 shows that how an overlay network creates a logical network over the existing physical underlay network. The virtual network is created using virtual routers, switches or logical tunnels whereas the underlay network is created using Ethernet, MPLS or IP Network.

Each instance of virtual network is implemented as an overlay network.

a. The packet will be encapsulated at the edge device at the first hop called NVE and will be send to the remote NVE.

b. The encapsulation done at the edge device,NVE will determine the destination point of the edge device at which decapsulation is to be performed in the other side of network to the before delivering to the VM.

c. The packet is forwarded based on the encapsulated header in the network.

4.3.1 Control Plane:Overlay Networks

The three areas in the control plane protocol which needs to realize an overlay solution:-

a. It needs to maintain the mapping tables. This is done by Network Virtualization Authority(NVA) which is responsible for maintaining and distributing the mapping information to the entire overlay system.

b. The attachment and detachment of VMs from aspecific virtual network instance.

c. There should be one standard protocol for the NVA-NVE interaction.

4.3.2 Data Plane:Overlay Networks

The encapsulated packet is carried by the data plane for the tenant systems. The VNI(Virtual Network Identifier) is present in the encapsulated header which identifies the packet belongs to which virtual network.

4.4 Summary

This chapter describes the data center architecture and requirements of modern data center network. It also describes about overlay networks, control plane and data plane operations for this networks.

Chapter 5

VXLAN and NVGRE - Layer 2 Over Layer 3 Protocols

5.1 Why Layer-2 Overlay Protocols

a. Limitations due to STP and VLAN:-

To avoid frame replication and frame looping, STP are used in the Layer 2 networks. Some of the ports and links gets blocked after the network is configured with STP. It seems that we are paying more for ports and links from which we can really use.

VLANs have been extensively used to provide broadcast isolation in the Layer 2 data center networks. A maximum of 4094 VLANs can be used since a VLAN Id is of 12-bit. But due to recent expansion of data center networks and use of STP, this number seems to be inadequate.

b. Multitenant Environment:-

In multi-tenant environments, Cloud computing requires on demand allocation of resources. Layer 2 networks such as VLANs are used to segregate the network traffic for the tenant. So, VLAN Id is used to identify the tenant in the network. But cloud provider might provide more number of tenants, the

4094 VLAN limit seems to be inadequate. Also it may be possible that same tenant might be using the multiple VLANs, which worsen the issure more.

c. Inadequate Table Sizes at ToR Switch:-
In virtualized environments several VM may be connected to the single port of the switch. So, Switch has to learn the MAC address of multiple VMs on that port. These requires large capacity of forwarding tables compared to non-virtualized environments. If the table overflows, it may stop learning the new addresses until idle entries age out, leading to the flooding of subsequent unknown destination frames.

5.2 VXLAN-Virtual eXtensible LAN

The goal of VXLAN is to provide seamless Layer 2 connectivity and isolation to a tenant's resources across a Layer 3 data center network. It is created using a "MAC inside UDP" encapsulation scheme. The transport protocol over the physical data center network is IP+UDP. Each overlay is known as VXLAN segment. The VXLAN Network Identifier(VNI), which is of 24-bit is used to identify the VXLAN segment. Therefore maximum of 16 M VXLAN segments can exists under a common administrative domain. The VMs having the same VNI can only communicate with each other. Virtual machines are identified by MAC+VNI. So, it is possible to have duplicate MAC addresses in N different VXLAN segments without issue but not in the same VXLAN segment.
The two different segments can have two VMs which have identical MAC addresses but the traffic cross over would never occur since they are isolated by the VNI associated with their own segment. Thus, MAC address will be encapsulated by VNI. Thus tunnels are stateless. The VTEP is located in the hypervisor on the server that hosts the VM. Thus all the encapsulating and decapsulating of header is only known to VTEP and VM never knows about it. The end point of the tunnel is located within the hypervisor on the server that hosts the VM. Thus, the VNI

and VXLAN-related tunnel / outer header encapsulation are known only to the VTEP and the VM never sees it. An overlay endpoint implemented in a hypervisor would typically only provide a mapping function - mapping VM traffic to a virtual network and vice-versa. It leaves reachability issues to the network infrastructure. An overlay endpoint implemented in a network switch necessarily implements both the mapping function and the data path forwarding functions.
The association of VM's MAC to VTEP's IP address is discovered via source-address learning. From an addressing perspective:-

a. Each overlay endpoint is identified by a unique IP address that identifies the endpoint and describes its location within the physical network topology.

b. The originating and terminating overlay endpoints can be in the same data center or can span data centers. Their location is not limited by the VXLAN architecture but may be limited by practical management and scalability concerns.

   (1) Within each virtual tenant network, basic connectivity is provided at Layer-2. MAC addresses are assumed to be unique only within the scope of the virtual L2 network. Each virtual tenant network obviously implements its own IP addressing scheme but this is private to the virtual network and not visible to the data center core.

5.2.1 Unicast Communication

Let us assume a VM in VXLAN overlay network. This VM is unaware of the VXLAN network. To transmit a frame to a VM on a different VXLAN network, it sends a MAC frame destined towards the target VM as normal frame as if it is doing normal Layer 2 switching. The VTEP present on that physical host determines that VNI to which it belongs. If it is on the same physical network then it does the switching normally. If it is on the different physical network then there is

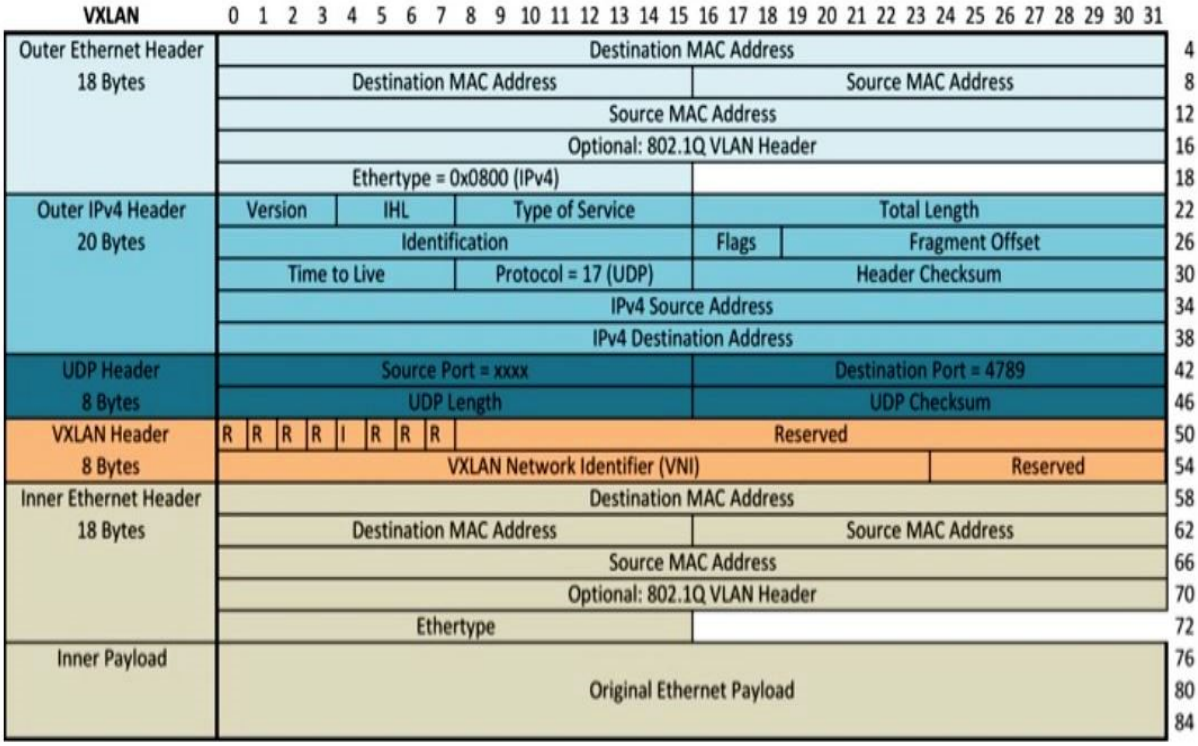

Figure 5.1: VXLAN Frame Format

mapping between DA to the remote VTEP. This packet is encapsulated at VTEP with VXLAN header, IP header and MAC header and forwarded towards remote VTEP. The received packet at remote VTEP is validated by VTEP that DA of VM on that VNI matches inner DA. If so, the packet is decapsulated and passed on to its destination VM.

During this process, the remote VTEP learns the mapping between the inner SA to outer IP address so that when this VM wants to respond, no flooding mechanism is required.

5.2.2 Broadcast and Multicast Communication

Consider the VM on the source host attempting to communicate with the destination VM using IP. Assuming that they are both on the same subnet, the VM sends out an Address Resolution Protocol (ARP) broadcast frame. In the non-VXLAN environment, this frame would be sent out using MAC broadcast across all switches carrying that VLAN.

With VXLAN, a header including the VXLAN VNI is inserted at the beginning of the packet along with the IP header and UDP header. However, this broadcast packet is sent out to the IP multicast group on which that VXLAN overlay network is realized. To effect this, we need to have a mapping between the VXLAN VNI and the IP multicast group that it will use. This mapping is done at the management layer and provided to the individual VTEPs through a management channel.

The destination VM sends a standard ARP response using IP unicast. This frame will be encapsulated back to the VTEP connecting the originating VM using IP unicast VXLAN encapsulation. This is possible since the mapping of the ARP response's destination MAC to the VXLAN tunnel end point IP was learned earlier through the ARP request.

5.2.3 VXLAN Deployment Scenario

In data centers VXLAN is typically deployed on virtualized hosts, which may be present across mulitple racks. This individual racks can be part of either single Layer 2 network or different Layer 3 network. The VXLAN segments/overlay networks are overlaid on top of these Layer 2 or Layer 3 networks.

Consider Figure 5.2, which depicts two virtualized network attached to a Layer 3 infrastructure. The network could be on the same rack, on different racks, or potentially across data centers within the same administrative domain. There are four VXLAN overlay networks identified by the VNIs 11, 22, 33 and 44. Consider

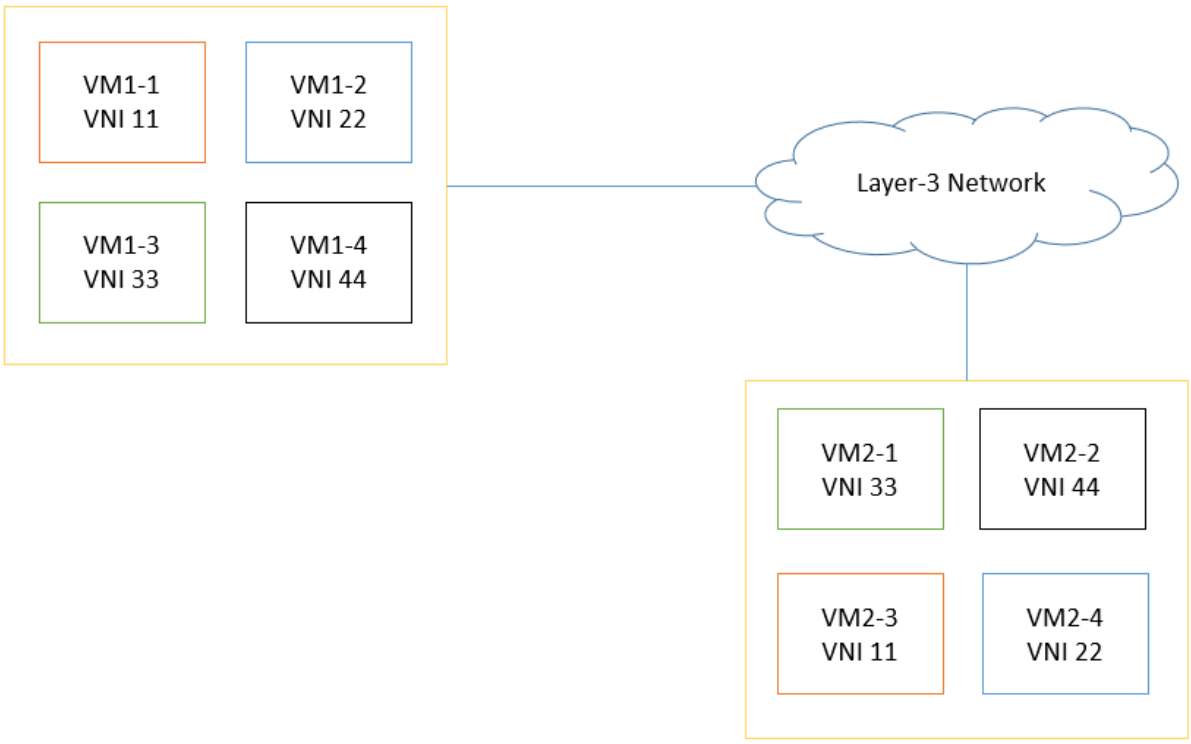

Figure 5.2: VXLAN Deployment Scenario

the case of VNI 11 which is VM1-1 on server 1 and VM2-3 on server 2. Since the encapsulation and decapsulation happens at the VTEPs of server 1 and 2, these VMs doesn't know about the overlay networks and transparent method. Similarly VNI 22 identifies VM1-2 on server 1 and VM2-4 on server 2, VNI 33 identifies VM1-3 on server 1 and VM2-1 on server 2, VNI 44 identifies VM1-4 on server 1 and VM2-2 on server 2.

5.3 NVGRE-Network Virtualisation using Generic Routing Encapsulation

Network virtualization enables multiple server instances to run concurrently an a single physical host and yet services instances are isolated from each other. It involves creating virtual Layer 2 topologies on top of a physical Layer 3 network. The virtual topology gets the connectivity by tunneling Ethernet frames in GRE over the physical network. Each VM network consists of one or more virtual subnets. A VM network forms an isolation boundary where the virtual machines within a VM network can communicate with each other. As a result, virtual subnets in the same VM network must not use overlapping IP address prefixes.

In NVGRE, the virtual machines packet is encapsulated inside another packet. The header of this new packet has the appropriate source and destination PA IP address in addition to the Virtual Subnet Identifier(VSID) which is stored in key field of the GRE. The unique identification of a tenant's virtual subnet is done by VSID which is associated with every virtual Layer-2 network carried in the outer header. The VSID is of 24-bits which helps us to support upto 16 million virtual subnets as compared to 4K VLANs. Each VSID corresponds to virtual Layer 2 broadcast domain. Any protocol can be encapsulated over an IP protocol using GRE header. The VSID information is carried in the GRE header for the NVGRE protocol.

The encapsulation and decapsulation of the overlay header happens at the point known as NVGRE endpoint. This points are placed in betweeen the virtual and the physical networks. This endpoint can optionally take part in routing and functions as a gateway in the virtual topology. The encapsulations header can be determined by control plane and data plane or by the management plane.

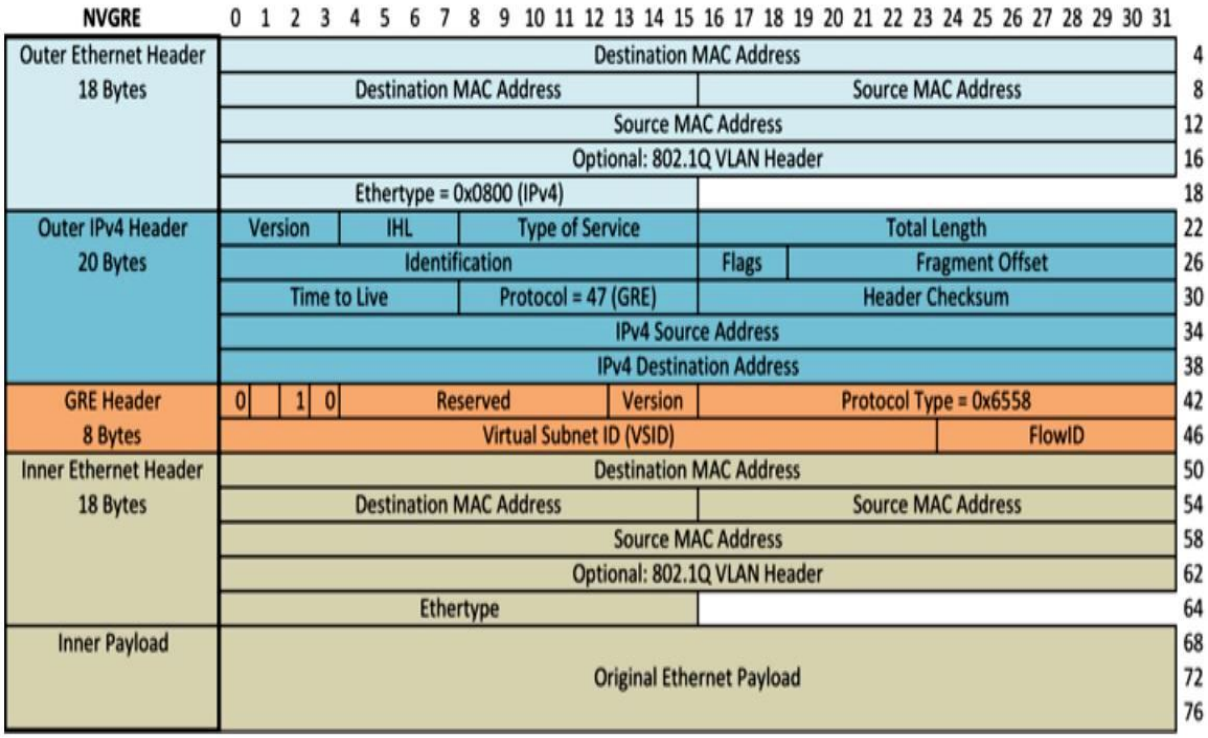

Figure 5.3: NVGRE Frame Format

5.3.1 Unicast Communication

The encapsulation is done at the NVGRE endpoint on a Layer-2 packet in GRE with the appropriate source PA and destination PA. There may be chance that multiple PAs are associated with an endpoint, in that case policy will determine which will be used. The encapsulated packet is switched normally to the destination PA. At the destination, NVGRE endpoint decapsulates the GRE packet and teh packet is switced using inner destination address.

5.3.2 Broadcast and Multicast Communication

The Broadcast and Multicast communication is the same way as N-way unicast. The packet will be encapsulated at one NVE N times and sent to each NVE in the virtual subnet.

5.4 Summary

This chapter describes the Layer-2 over Layer-3 protocols such VXLAN,NVGRE.It also discusses about the frame format, Unicast,Multicast and Broadcast Communication for this protocols.

	VXLAN	NVGRE
Encapsulation	* Uses UDP based encapsulation. * Uses UDP port 8472. * Adds an 8-byte VXLAN header.	* Uses GRE based encapsulation. * Uses GRE protocol type 0x6558 (Transparent Ethernet Bridging)
Overlay Identification	24-bit Virtual Network Id(VNI).	24-bit Virtual Subnet Id(VSID), plus an optional 8-bit Flow-Id.
Encapsulation overhead	50 Bytes	42 Bytes
Forwarding of Layer-2 Broadcast,Multicast and Unknown Unicast traffic	* Encapsulation uses IP multicast as destination IP. * Each VNI is mapped to multicast group. * Multiple VNIs can share same multicast group	* Encapsulation uses IP multicast as destination IP. * Each VSID is mapped to multicast group. * Multiple VSIDs can share same multicast group.
Address learning and Control plane	* Learning and Flooding approach i.e, Data plane based learning. * Separate Control plane.	* We can use any,mechanism to distribute,location and VSID,information:data plane, learning,control-plane,based, etc.

Chapter 6

Test Scenarios

This chapter summaries the scenarios of the various features like Layer-2 switching, Layer-3 Switching, IP Tunneling, VXLAN and NVGRE that were being validated. Also for each of these features, various packet formats are shown which depicts the headers associated with them.

Table 6.1 shows the test scenarios for Layer-2 switching.
Table 6.2 shows the test scenarios for Layer-3 switching.
Table 6.3 shows the test scenarios for IP Tunneling.
Table 6.4 shows the test scenarios for VXLAN.
Table 6.5 shows the test scenarios for NVGRE.

Table 6.1: Test Scenarios for Layer-2 Switching

Sr. No.	Test Scenarios
1	Layer-2 unknown source or destination unicast handling and self-learning
2	Hash-based lookup in layer-2 address table for every incoming packet
3	Support for PFM
4	Support for station movement
5	Support for CPU managed learning(CML)
6	Flooding of packets when there is a miss for the destination address in the hash table
7	Support for Layer-2 multicast
8	Support for Layer-2 broadcast
9	Support VLAN management per spanning tree
10	Classify received frames belonging to exactly one VLAN ID
11	Submit to the forwarding process and learning process,all the frames that are not discarded
12	Support for classification of VLANID based on port, protocol, MAC, IP-Subnet
13	Discard the frames if the transmission port is not present in the member set for frames VLAN ID
14	Support for changing the priority of the packet
15	Encoding of transmitted bridge protocol data units(BPDUs) and validating the received BPDUs
16	Configuring the ports of switch in accordance with the STP

Table 6.2: Test Scenarios for Layer-3 Switching

Sr. No.	Test Scenarios
1	Change the Destination MAC address,Source MAC address,VLAN ID while routing
2	Check for the ethertype values such as 0800 for IPv4 or 86DD for IPv6
3	Drop the packet if TTL is equal to zero
4	Decrement the value of TTL while routing
5	Drop the packet if the header checksum fails
6	Perform recomputation and verification of checksum after packet header processing at each point in internet
7	Choose a next-hop destination for each IP packet based on the information in its routing table
8	The Switch must use the longest prefix match algorithm while forwarding the traffic
9	Check for the minimum IP header length such as 20 bytes for IPv4 and 40 bytes for IPv6
10	Check that the switch discards the received packet which contains the IP destination address which is invalid

Table 6.3: Test Scenarios for IP Tunneling

Sr. No.	**Test Scenarios**
1	Check that when the switch is encapsulating a packet for IP tunneling, it doesn't change the inner IP header except to decrement the TTL
2	The total length measures the length of the entire encapsulated IP packet, including the outer IP header,the inner IP header and its payload
Note:	Test done for all IP Tunnels(4in4,4in6,6in4,6in6)

Table 6.4: Test Scenarios for VXLAN

Sr. No.	**Test Scenarios**
1	Check for encapsulation at VTEP for unicast packet
2	Check for decapsulation at VTEP for unicast packet
3	Check for encapsulation at VTEP for multicast packet
4	Check for decapsulation at VTEP for multicast packet

Table 6.5: Test Scenarios for NVGRE

Sr. No.	**Test Scenarios**
1	Check for encapsulation for unicast packet
2	Check for decapsulation for unicast packet
3	Check for encapsulation for multicast packet
4	Check for decapsulation for multicast packet

Transmitted Packet	Received Packet
00 01 02 03 04 05 10 01 02 03 04 05 08 00 45 38 00 6e 00 00 00 00 40 06 6a 48 90 0a 00 01 7c 00 03 ff 10 00 20 00 00 00 00 01 00 00 00 00 50 00 00 00 00 00 00 00 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 00 01 79 bd a5 d6	00 01 02 03 04 06 10 01 02 03 04 05 81 00 80 14 08 00 45 38 00 6e 00 00 00 00 40 06 6a 48 90 0a 00 01 7c 00 03 ff 10 00 20 00 00 00 00 01 00 00 00 00 50 00 00 00 00 00 00 00 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 00 04 da ea 50 b0

Figure 6.1: Result for Layer-2 Switching

Transmitted Packet	Received Packet
20 20 00 00 02 00 20 21 22 23 24 26 08 00 45 3a 00 6e 00 00 00 00 3f 09 64 34 0a 0a 01 01 0a 0a 02 05 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 00 04 f9 c2 28 d8	22 33 44 55 66 70 20 22 33 44 55 60 81 00 61 90 08 00 45 3a 00 6e 00 00 00 00 3e 09 65 39 0a 0a 01 01 0a 0a 02 05 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 00 01 d5 6b cc 20

Figure 6.2: Result for Layer-3 Switching(IPv4)

Transmitted Packet	Received Packet
20 20 00 00 02 00 20 21 22 23 24 26 86 dd 60 20 00 01 00 46 06 3f 00 00 00 01 00 02 00 03 00 04 00 00 00 00 00 00 00 18 00 01 00 02 00 00 00 04 00 05 00 06 03 f0 10 00 20 00 00 00 00 01 00 00 00 00 50 00 00 00 00 00 00 00 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 00 01 d2 97 18 28	22 33 44 55 66 70 20 22 33 44 55 60 81 00 61 90 86 dd 60 20 00 01 00 46 06 3e 00 00 00 01 00 02 00 03 00 04 00 00 00 00 00 00 00 18 00 01 00 02 00 00 00 04 00 05 00 06 03 f0 10 00 20 00 00 00 00 01 00 00 00 00 50 00 00 00 00 00 00 00 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 00 01 46 eb 1b 4c

Figure 6.3: Result for Layer-3 Switching(IPv6)

Transmitted Packet	Received Packet
10 02 03 04 05 01 20 21 22 23 24 01 08 00 45 fc 00 6e 00 00 00 00 3f 3b 48 28 0b 0b 0b 0b 0e 0e 0e 0e 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 00 01 1c 34 86 4d	00 11 22 33 44 55 50 55 55 55 55 55 81 00 a0 05 86 dd 61 40 00 00 00 6e 04 05 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00 05 00 00 00 01 00 02 00 03 00 04 00 05 00 06 00 05 45 fc 00 6e 00 00 00 00 3e 3b 49 28 0b 0b 0b 0b 0e 0e 0e 0e 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 00 01 de 1f dd 81

Figure 6.4: Result for IP Tunnel(ex:- 4in6 Encapsulation)

Transmitted Packet	Received Packet
10 02 03 04 05 01 20 21 22 23 24 01 86 dd 65 00 00 01 00 46 04 14 00 00 00 01 00 02 00 03 00 04 00 05 00 06 00 05 00 01 00 02 00 03 00 04 00 05 00 06 00 07 00 05 45 fd 00 46 00 00 00 00 40 06 47 84 0b 0b 0b 0b 0e 0e 0e 0e 10 00 20 00 00 00 00 01 00 00 00 00 50 00 00 00 00 00 00 00 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 00 01 3a c2 6b 42	00 11 22 33 44 55 50 55 55 55 55 55 81 00 a0 05 08 00 45 fc 00 46 00 00 00 00 3f 06 48 85 0b 0b 0b 0b 0e 0e 0e 0e 10 00 20 00 00 00 00 01 00 00 00 00 50 00 00 00 00 00 00 00 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 00 01 8d 19 49 93

Figure 6.5: Result for IP Tunnel(ex:- 4in6 Decapsulation)

Transmitted Packet	Received Packet
00 01 02 03 04 05 18 02 03 04 05 05 08 00 45 3a 00 6e 00 00 00 00 3f 09 6b 43 90 0a 00 01 7c 00 03 ff 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 00 01 49 02 db ff	22 33 44 55 66 77 20 22 33 44 55 60 81 00 e0 64 08 00 45 26 00 a0 00 00 00 00 ff 11 a4 12 0a 0a 01 01 0a 0a 02 00 ff ff 00 3e 00 8c 00 00 08 00 00 00 01 23 45 00 00 01 02 03 04 05 18 02 03 04 05 05 08 00 45 3a 00 6e 00 00 00 00 3f 09 6b 43 90 0a 00 01 7c 00 03 ff 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 00 01 bc 64 ec 87

Figure 6.6: Result for VXLAN Encapsulation

Transmitted Packet	Received Packet
20 20 00 00 02 00 20 21 22 23 24 26 08 00 45 3a 00 6e 00 00 00 00 3f 11 64 1e 0a 0a 01 0a 0a 0a 02 0a 00 3e 0f ff 00 5a 00 00 08 00 00 00 01 23 45 00 18 01 02 03 04 00 18 02 03 04 05 05 08 00 45 3a 00 3c 00 00 00 00 3f 09 6b 75 90 0a 00 01 7c 00 03 ff 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 00 01 68 21 40 ef	18 01 02 03 04 00 18 02 03 04 05 05 08 00 45 3a 00 3c 00 00 00 00 3f 09 6b 75 90 0a 00 01 7c 00 03 ff 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 00 01 99 ab ff f4

Figure 6.7: Result for VXLAN Decapsulation

Transmitted Packet	Received Packet
00 01 02 03 04 05 18 02 03 04 05 05 08 00 45 3a 00 6e 00 00 00 00 3f 09 6b 43 90 0a 00 01 7c 00 03 ff 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 00 01 49 02 db ff	22 33 44 55 66 77 20 22 33 44 55 60 81 00 e0 64 08 00 45 26 00 98 00 00 00 00 ff 2f a3 fc 0a 0a 01 01 0a 0a 02 00 20 00 65 58 12 34 50 00 00 01 02 03 04 05 18 02 03 04 05 05 08 00 45 3a 00 6e 00 00 00 00 3f 09 6b 43 90 0a 00 01 7c 00 03 ff 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f 50 51 52 53 54 55 56 57 00 01 bb a6 88 59

Figure 6.8: Result for NVGRE Encapsulation

Transmitted Packet	Received Packet
20 20 00 00 02 00 20 21 22 23 24 26 08 00 45 3a 00 6e 00 00 00 00 3f 2f 64 00 0a 0a 01 0a 0a 0a 02 0a 20 00 65 58 00 01 23 45 18 01 02 03 04 00 18 02 03 04 05 05 08 00 45 3a 00 44 00 00 00 00 3f 09 6b 6d 90 0a 00 01 7c 00 03 ff 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 00 01 60 5f 98 8b	18 01 02 03 04 00 18 02 03 04 05 05 08 00 45 3a 00 44 00 00 00 00 3f 09 6b 6d 90 0a 00 01 7c 00 03 ff 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 00 01 97 d4 38 8e

Figure 6.9: Result for NVGRE Decapsulation

Transmitted Packet	Received Packet
00 01 02 03 04 05 00 a0 b0 c0 d0 01 88 88 00 00 01 23 00 00 91 00 a0 14 08 00 45 3a 00 62 00 00 00 00 3f 06 60 47 0c 0a 01 01 0c 0a 02 01 0b b8 0f a0 00 00 00 01 00 00 00 00 50 00 00 00 00 00 00 00 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 00 01 f5 e5 02 9d	00 00 00 00 00 01 00 00 00 00 00 02 81 00 e1 90 08 00 45 26 00 90 00 00 00 00 ff 2f a4 02 0a 0a 01 01 0b 0b 01 01 20 00 65 58 11 11 11 11 00 01 02 03 04 05 00 a0 b0 c0 d0 01 91 00 a0 14 08 00 45 3a 00 62 00 00 00 00 3f 06 60 47 0c 0a 01 01 0c 0a 02 01 0b b8 0f a0 00 00 00 01 00 00 00 00 50 00 00 00 00 00 00 00 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 36 37 00 01 d3 f4 4d e1

Figure 6.10: Result for NVGRE with IEEE 802.1BR Encapsulation

Transmitted Packet	Received Packet
00 11 aa aa aa aa 00 a0 b0 c0 d0 01 08 00 45 26 00 6e 00 00 00 00 ff 2f a4 24 0b 0b 01 01 0a 0a 01 01 20 00 65 58 22 22 22 22 00 11 aa 00 00 01 00 11 aa 11 00 00 08 00 45 3a 00 44 00 00 00 00 3f 06 60 65 0c 0a 01 01 0c 0a 02 01 0b b8 0f a0 00 00 00 01 00 00 00 00 50 00 00 00 00 00 00 00 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 00 01 81 3b 95 69	00 11 aa 00 00 01 00 11 aa 11 00 00 88 88 90 00 01 23 00 00 08 00 45 3a 00 44 00 00 00 00 3f 06 60 65 0c 0a 01 01 0c 0a 02 01 0b b8 0f a0 00 00 00 01 00 00 00 00 50 00 00 00 00 00 00 00 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 18 19 00 01 6c 70 47 15

Figure 6.11: Result for NVGRE with IEEE 802.1BR Decapsulation

Chapter 7

Conclusion and Future work

7.1 Conclusion

Electronic System Level (ESL) validation is one of the most efficient and time saving methods of validating a chip during its design cycle. The ESL validation process must be included in the product design flow of highly complex SoC systems, such as high speed multilayer Ethernet switches and next generation Ethernet switches, as the industry requirement for better quality products with lesser time to market. Broadcom's High-Capacity StrataXGS Ethernet switch series will be mainly used in internet backbones and in data centers for large bandwidth and faster connectivity. It is highly dependent on secure Layer-2 overlay network features to enable reliable and secure communication of voice, video and data traffic on a single high performance network.

Tests have been carried out for Layer-2 (unicast, multicast and broadcast), Layer-3 (unicast, multicast and broadcast), IP Tunneling (encapsulation and decapsulation), VXLAN (encapsulation and decapsulation) and NVGRE (encapsulation and decapsulation). The bugs found were reported back to the implementation team for a fix. A bug free software model of the switch was delivered to RTL team for RTL implementation.

7.2 Future work

As the features and enhancements associated with the Data Link Layer and Network Layer are increasing day by day, therefore the future work is to deal with the challenges involved in ESL validation of new features of the Ethenet switch. The other features like STT (Stateless Transport Tunneling), GENEVE (Generic Network Virtualization Encapsulation) of the Ethernet Switch may be taken up in future.

References

[1] Andreas Gerstlauer,Christian Haubelt,Andy D. Pimentel,Todor P. Stefanov,Daniel D. Gajski,Jurgen Teich,"Electronic System Level Synthesis Methodologies" IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems, vol.28,no.10,October 2009.

[2] Shukla, S.K., Pixley, C., Smith, G. "Guest Editor's Introduction: The True State of the Art of ESL Design" Design and Test of Computers, IEEE, vol. 23, pp. 335-337, May 2006.

[3] Kogel, T., Takach, A., Martin, G., Donlin, A., Chatha, K."From ESL 2010 to ESL 2015" in proc Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2010 IEEE/ACM/IFIP International Conference, pp. 61-62, Oct 2010.

[4] Coussy, P."High Level Synthesis: On the path to ESL Design" in proc ASIC(ASICON)2011 IEEE 9th International Conference, pp. 1098-1101, Oct 2011.

[5] Weiwei Chen, Xu Han, Rainer Domer, "ESL Design and Multi-Core Validation using the System-on-Chip Environment", Center for Embedded Computer Systems, University of California, Irvine, USA.

[6] Rich Seifert, Jim Edwards "The Switch Book:The Complete Guide to LAN Switching Technology", Second Edition, Wiley Publications.

[7] Douglas E Comer, "Internetworking with TCP/IP", PHI Publications.

[8] "Data Center Overlay Technologies",White Paper,Cisco,2013.

[9] VXLAN: A Framework for overlaying Virtualized Layer 2 Networks over Layer 3 Networks,RFC 7348,2014.

[10] NVGRE:Network Virtualization using Genric Routing Encapsulation,draft-sridharan-virtualization-nvgre-07.txt,2014.

[11] Problem Statement: Overlays for Network Virtualization,draft-ietf-nvo3-overlay-problem-statement-04,2013.

[12] IP in IP Tunneling,RFC 1853,1995.