ELECTRONIC SYSTEM LEVEL VALIDATION OF ETHERNET SWITCH

Submitted By

Jaydip Kansara

13mcec07

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

April 2015

ELECTRONIC SYSTEM LEVEL VALIDATION OF ETHERNET SWITCH

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering

Submitted By

Jaydip Kansara

(13mcec07)

Guided By

Prof. Rupal Kapdi

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

April 2015

Certificate

This is to certify that the major project entitled **"ELECTRONIC SYSTEM LEVEL VALIDATION OF ETHERNET SWITCH"** submitted by **Jaydip Kansara (Roll No: 13MCEC07)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering of Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this project, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Prof. Rupal Kapdi
Guide & Assistant Professor,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Prof. Vijay Ukani
Associate Professor,
Coordinator M.Tech - CSE
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sanjay Garg
Professor and Head,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr K Kotecha
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Jaydip Kansara**, Roll. No. **13MCEC07**, give undertaking that the Major Project entitled "**ELECTRONIC SYSTEM LEVEL VALIDATION OF ETHERNET SWITCH**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made. It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Signature of Student

Date:

Place:

Endorsed by

Prof. RupalKapdi

(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof. Rupal Kapdi**, Assistant Professor, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr K Kotecha**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

See that you acknowledge each one who have helped you in the project directly or indirectly.

- Jaydip Kansara

13MCEC07

Abstract

The project ESL Validation of Ethernet Switch aims at validating the multiple features of the switch used in the data center environment.These features include the data link layer, network layer and overylay network features of Ethernet switch. Results are compared with the expected result of the correct behavior of the switch.Bugs which are found during the validation are reported back to the software model of the switch to obtain the bug free Register Transfer Level(RTL) implementation.

Abbreviations

ESL	Electronic System Level Validation.
VLAN	Virtual Local Area Network.
VxLAN	Virtual Extenisible Local Area Network.
SOC	System On Chip
RTL	Register Transfer Level
CAM	Content Addressable Memory
VNI	VxLAN Network Identifier
VTEP	VxLAN Tunnel Endpoint

Contents

List of Figures

Chapter 1

Introduction

1.1 Electronic System Level Validation

1.1.1 What is Electronic System Level Validation(ESL)?

When the compleity of the System On Chip starts to grow, designers face difficulties.If we raise the level of abstratciton then this complexity can be handled in a better way. ESL is one kind of high level language which is based on the design flow. We need efficient ESL framework which can transform and ultimately refine the high level design model. Validation is very critical and complex activity in the current System on Chip (SoC) design process. As chip design complexity grows, validation methodologies become more complicated in the current SoC design process. Instead of prolonging the process of finding bugs in register transfer level (RTL) code, the design flow is geared towards creating bug-free RTL designs. This is realized by automating the generation of RTL from exhaustively verified system software models using Electronic System Level (ESL) validation. ESL validation can capture errors that occur at the very early stage of chip design cycle.Due to this early capture in design model, lot of cost can be saved in terms of money and time. Basically in ESL, we convert architecture specification into a chip model based on SystemC programming. ESL validation of the network switch covers only the functional aspect of the chip; it does not address performance aspect.

1.1.2 ESL and EDA

ESL validation captures errors that occur during the architectural design phase of a chip. The bugs found in the ESL validation are very fundamental to the design. EDA comprises

of:

- Synthesis Processitem
- Hand-in-hand design og hardware and software
- Verification Tasks
- Test tools used for checking ESL design
- Translation of a correct ESL design to a RTL
- Design at the Gate level
- Design at the switch level
- production the physical design described in the GDS-II format which is ready for fabrication and manufacturing

EDA is kind of a combination of test automation and Design automation tools that takes care of automation of the design procedures and test procedures.

- Design Automatio Tools
- Test automatio tools : Electronic System's quality is dealt at this level of stage.

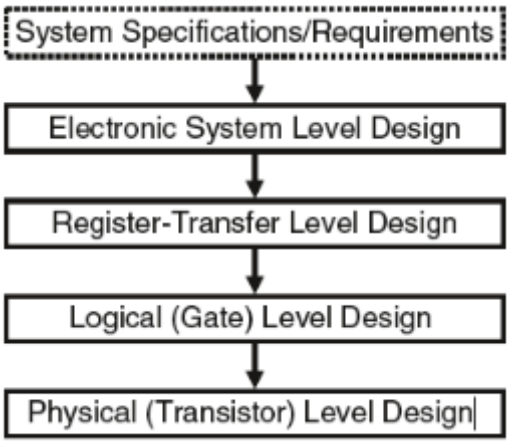

Figure 1.1: Typical Chip Design Flow

-

1.1.3 Design Hierarchy

ESL design partitions the systems into hardware and software system design. Co-simulation and co-design is takes care at ESL. Using ESL, it is very easy to have cost-estimation done at early stage. Design-space exploration can also be done at this stage. When higher level description is seen at the design level, it can be seen that it has less implementation details.But advantage here is that it has more functional information than any lower level details.

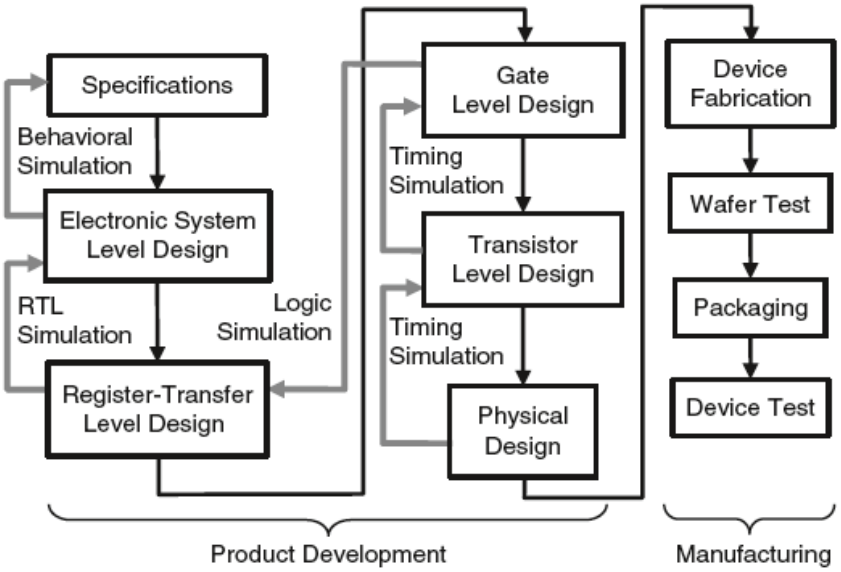

Figure 1.2: IC Design and Verification Flow

-

1.1.4 ESL and functional verification of system-on-chip

Modern verification environment are built based on the object oriented concepts. We can have system-level solution for the SoC.In this model all the data/processes can be managed.

These type of systems proved us with integration of smaller blocks into larger block. Eventually these smaller blocks can be combined to make a larger and integrated system. Due to this purpose,design and functional verification can be done at a system level.Because of this, teams can create larger and more powerful SoC rapidly. ESL can describe the SoC design in a very abstracted details. This can serve the purpose of the extraction of the design space.ESL also provides the virtual implementation for software and hardware implementation details.

Process of developing software can start early if given the virtual prototype.Embedded software can be written and tested by multiple software engineering groups in parallel because model of the hardware is available with us. This can save a lot of effort and time in the development. Hardware and software verification procedure can be made more clearer and simpler because of this.

Functional verification of SoC indicates that the design implementation complies the design specification. It is contained of :

- Stimulus events and results which are expected, which basically verifies that design generates the correct results when given the stimulus events
- Golden plan and stimulus event constraints, which verifies that the golden plan and RTL level behavior are coherent.

Functional verification based on ESL is composed of 4 phase :

- Having the appropriate stimuli, test scenarios are developed
- Test cases are exucuted on the TLM model test bench to have the final Golder reference model
- Test cases are executed on the DUT to examine the functional verification of the DUT.
- Comparison of output results with the expected/correct behavior

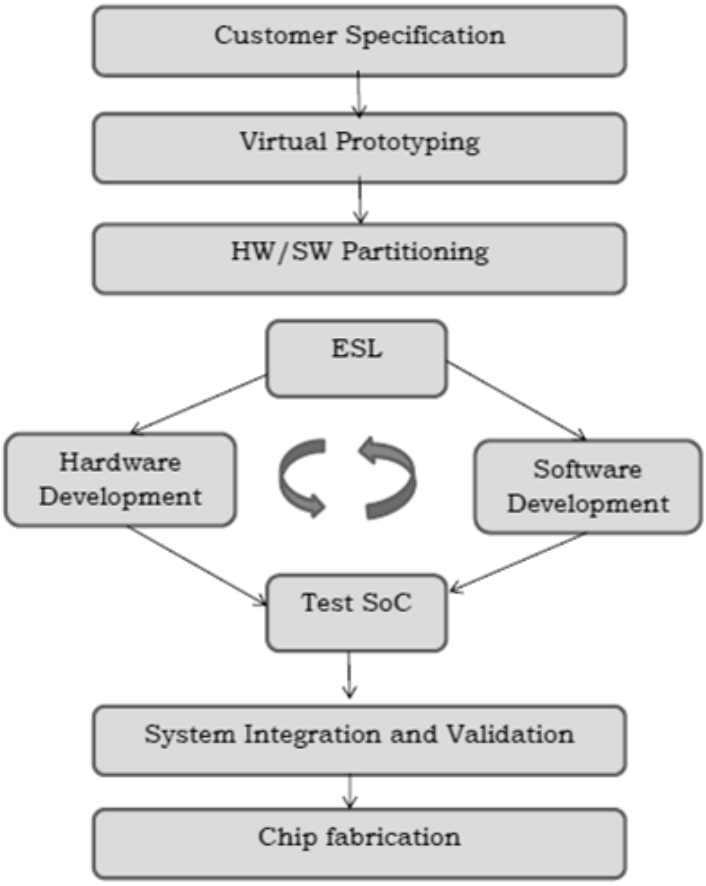

Figure 1.3: ESL and functional verification of SoC

-

1.1.5 ESL Validation model

1. Test Case:-

   TCL scripting language is used for validating the s/w model of the ethernet switching chip. Scripts are written in TCL for test cases. Unix platform is used for the execution of test cases. Test cases consist of three parts : Transmitted packet configurations, expected packet configurations and configuratinos programmed in the interface file.

2. Packet Generator:-

   TCL scripts provides the data for packet generator. Packet generator processes the parameter and procuded appropriate packets.It is interfaced with the TCL for test generation.The interface is responsible for the input and monitoring the software model's response. Basically packet generator is utilized for generating network

traffic in the validation environment.

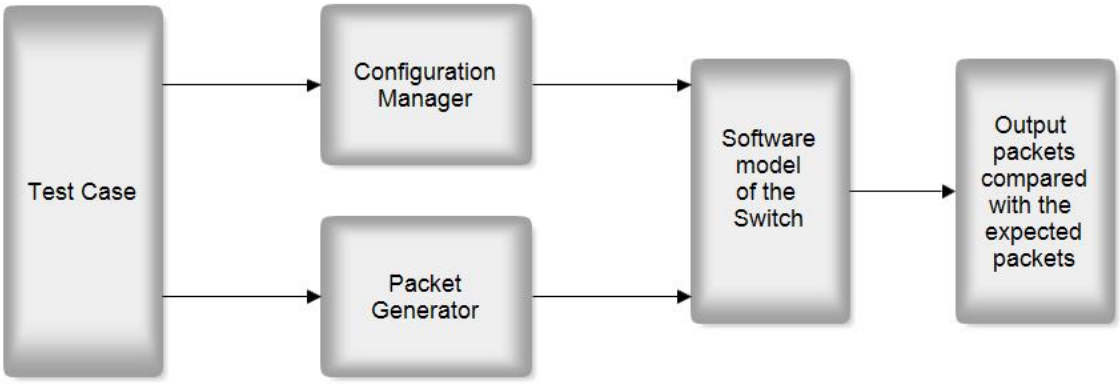

Figure 1.4: ESL Validation model

-

1. Configuration Manager:-

   It facilitates all the information needed to configure the software model of the ethernet switching chip. For validation, test cases along with the configuration information are provided to the validation environment.

2. Software Model of the Switch:-

   Arch Specs defined for the product is excuded in a software model. It is written in a language called System C. This System C model is given the test cases and configuration programmed by user. By giving the test cases, we validate the System C model.

Chapter 2

Layer 2 Switching

2.1 Layer-2 Switching

Layer-2 switch functionalty includes the analysis of incoming frames. It then forwards the frame based on the data contained in the frame.It basically sees the packet and then based on the packet, decides to forward/drop or flood the packet.Switches observes the destination MAC addresses and takes forwarding decisions. In Layer-2, switch performs the following functions.:

1. Dynamic learning of MAC SA:

   Ethernet switch dynamically parse the packet for the MAC SA and it stores the MAC SA in the CAM memory. Storing mechanism is the mapping of that MAC SA with the VLAN id and the incoming port number. So now when a port on that switch receives any packet, it parses the packet and sees the MAC DA. Now it checks that MAC DA in that CAM table, if match happens for that DA, it will directly send the packet to that particular port only. This CAM table can not be kept as it is, because some host might have moved from one location to another(station movement). Some stale entries are needed to be removed from the table. This is achieved through a mechanism called "aging". If a switch does not listen for a particular time from a particular machine/host, it removes that entry from the table.

2. Frame forwarding:

   The switch parses the packet, and observes the MAC DA, then it observes the MAC SA. It makes entry for this MAC SA in the CAM table. Then it transfers the packet

to desired port number based on the lookup done in the CAM table. If there is a miss in the table for that MAC DA, then it floods the packet to the incoming VLAN.

2.2 Virtual Local Area Network(VLAN)

Virtual LAN (VLAN) technology partitions the physical segments of the single LAN.It is a logical separation of the segments.All the hosts are although connected via physical cables which ultimately lead to physical wiring devices. .All the hosts and applications can communicate as seamlessly as if they were on a single LAN and all those hosts can be controlled configuring the switch. The LAN is virtual meaning that all the applications and hosts can be treated as if they are connected to a single physical LAN, but actually they are not. VLANS allows us to logically group the hosts in single broadcast domain. Hosts can be grouped in one common group, and because of this they are only able to communicate with the hosts in the same group and not outside that group.

2.2.1 VLAN assignment strategies

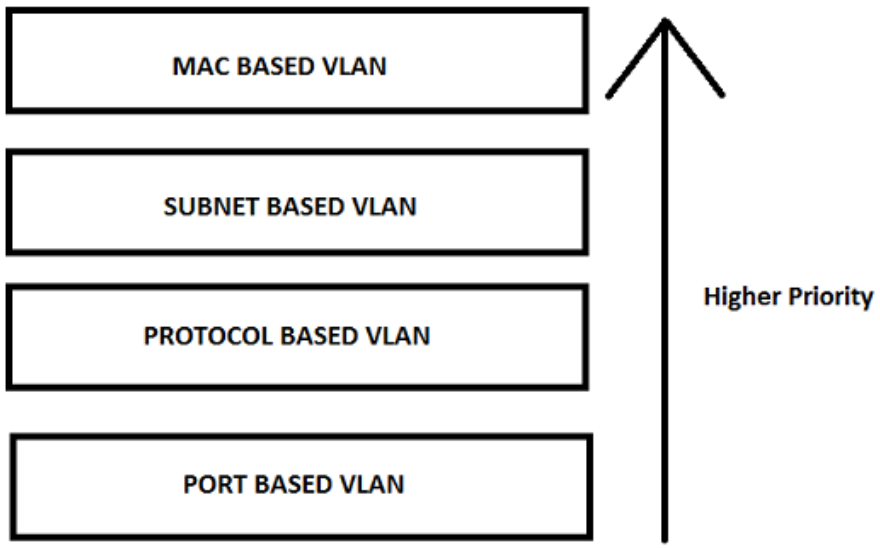

Figure 2.1: VLAN assignment techniques and priority

-

1. Port Based VLAN Mapping:

2. MAC Based VLAN Mapping:

3. Protocol Based VLAN Mapping:

4. IP Subnet Based VLAN Mapping:

Precedence can be changed among MAC based and Subnet based assignment by setting the precedence value as 1 or 0 in the register. If any of these tables are not programmed, then by default VLAN id is picked from the PORT table.

Chapter 3

Data Center

3.1 Data center concepts

Before the concept of data centers, all the storage elements, computing elements and the interconnecting network used to be on the desktop Personal Computers of the organization.But with time, data started to grow in size and with the increased data, to handle these volume of data, departmental servers were introduced. But departmental servers could not meet the need of collaboration among users. To meet this requirement, more centralized mechanism was introduced as data centers. Because of data centers, software and hardware management became easier. Data centers provided more efficient way to maintain it. As it was centralized, all users shared information in a more efficient manner.

Basically, data centers were created in order to separate the storage elements with the computing elements and the network that connected them with the users/clients. A shift happened for data center technology when VMWare introduced virtualization. They introduced the new technology that allowed the native operating system to execute one more other operating system just as if they were just another software installed in the operating system. In order to achieve it, they created one virtual environment that gave an impression of real computing system. They allocated resources to virtual machines. This supervisor program is called hypervisor.

As and when advancements in the memory, storage and computing elements happened, data centers became more capable of running more and more operating systems in the virtualized environment. Network administrators had the capability to dynamically scale the resource requirement for each operating system. This is called elastic

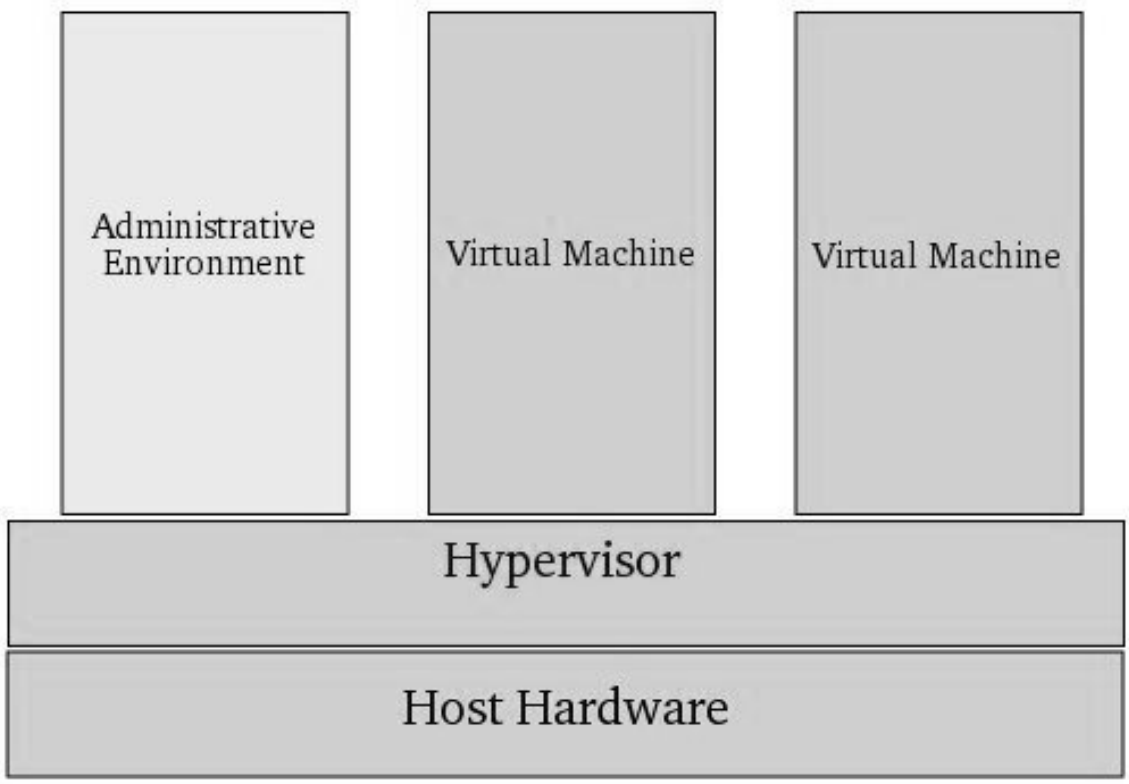

Figure 3.1: Hypervisor

-

computing.

With the introduction of elastic computing, it was very easy to move any virtual machine from one physical server to another physical server located in another location. Admins just had to pause the virtual machine and move to another physical server.This flexibility allowed the network administrators to have proper resource utilization and allocation.

Along with all the advantages, this new technology brought one issue for network operators. As and when network operator anticipated the growing demands, they would order all the equipment that would be needed to fulfill the requirement for the following year. After having these equipments, they put it on the racks. They would consume power even if they are not being used for long time. Because of these cooling instruments were also required to have the temperature as per the need.

Amazon was the first company to discover this problem. Amazon's business was growing very rapidly. It was doubling almost every 6 to 9 months. They anticipated the

requirement and pre-purchase all the equipment in order to meet the growing need. So that is where they identified the problem because machines used sit idle for very long time. This is when Amazon Web Service(AWS) was introduced. The idea behind AWS was to still pre order the storage and computing elements, but instead of having them sit idle they thought of leveraging those resources by selling them to other companies which needed these kind of resources.

1. **Core Layer** :Basically it Provides the packet switching backplane for every flows which goes in and out of the data center.Router or switch at the core layer runs an interior routing protocol like OSPF or EIGRP

2. **Aggregation layer** : composed of high-capacity core nodes (Ethernet switches/routers). Core nodes may provide virtual Ethernet bridging and/or IP routing services.

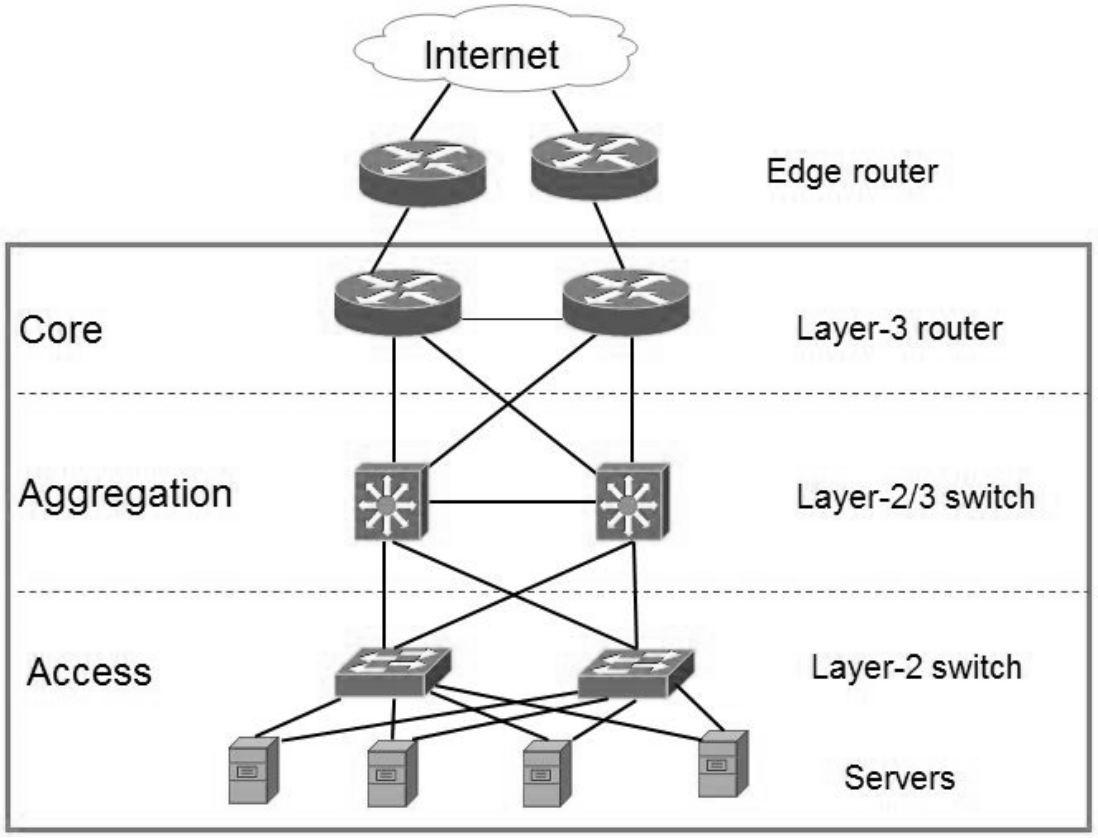

Figure 3.2: Data center architecture

-

1. **Access layer** :Physically connects servers to the network.Each server is connected to two access switch. typical example of an access switch is a Top-of-Rack (ToR)

switch. Other deployment scenarios may use an intermediate Blade Switch before the ToR, or an End-of-Row (EoR) switch, to provide similar functions to a ToR.

-

3.2 Multitenant Data Centers

Amazon Web Service(AWS) allowed different customers to let in their networks. This introduced the new requirement for them, how to separate thousands of tenants whose resources need to be spread across different physical location in different physical servers. Even after ensuring this, they need to give seamless access of resources in the Amazon Cloud.

The concept of tenant is not tied to any specific definition. It can mean different in different situation. For example, in a service provider data center, tenant can mean a single customer. In an enterprise data center, tenant can mean a department of the organization.

The network architects introduced a model of multitenancy. In this model :

- The IaaS(Infrastructure as a Service) and PaaS(Platform as a Service) requires highest degree of multitenancy.
- When SaaS is single tenant, it requires lowest degree of multitenancy.

3.3 Virtualized Multitenant Data Center

Virtualized multitenant data center allows data centers to host multiple tenants and provide them with the virtual slice of resources. Typically each tenant is a set of virtual machines. All these virtual machines are hosted on servers which runs hypervisors.These hypervisors contains virtual switches(vSwitch). These virtual switches connect all the virtual machines to each other and to the physical network.

Servers in the data centers are connected using high speed Ethernet network. Basically, it is layer-2 network. For overlay kind of solutions, data centers might be layer 3 network like IP, MPLS or GRE.

To provide isolation for each tenant, they are given one private network. In this network, all the virtual machines can communicate with each other. But they cannot communicate with the virtual machine of the other tenant.

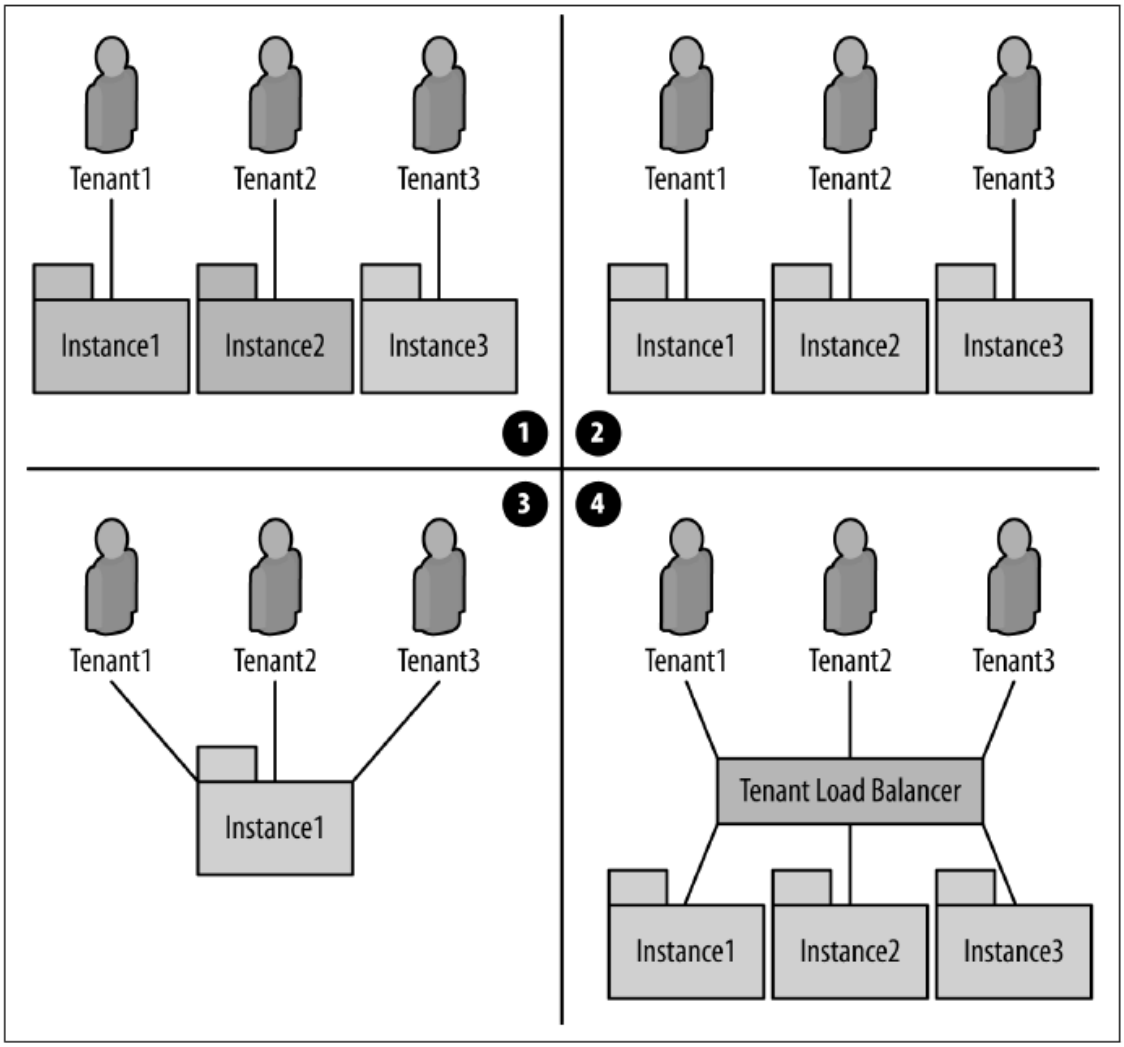

Figure 3.3: Multitenant data center

-

Typically tenant's network is a layer-2 network. All the virtual machines in the particular tenant are given the same layer 3 IP subnet address. IP address need not be unique in different tenant's network.

3.4 VxLAN :Virtual Extensible LAN

3.4.1 Server virtualization and VxLAN

A physical server now has multiple virtual machines running in it.With the rapid development of cloud computing and cloud service providers, their data center typically include more than one tenant. This kind of data centers are called multitenant data centers. These data centers basically has lots of physical servers and inside each physical server,

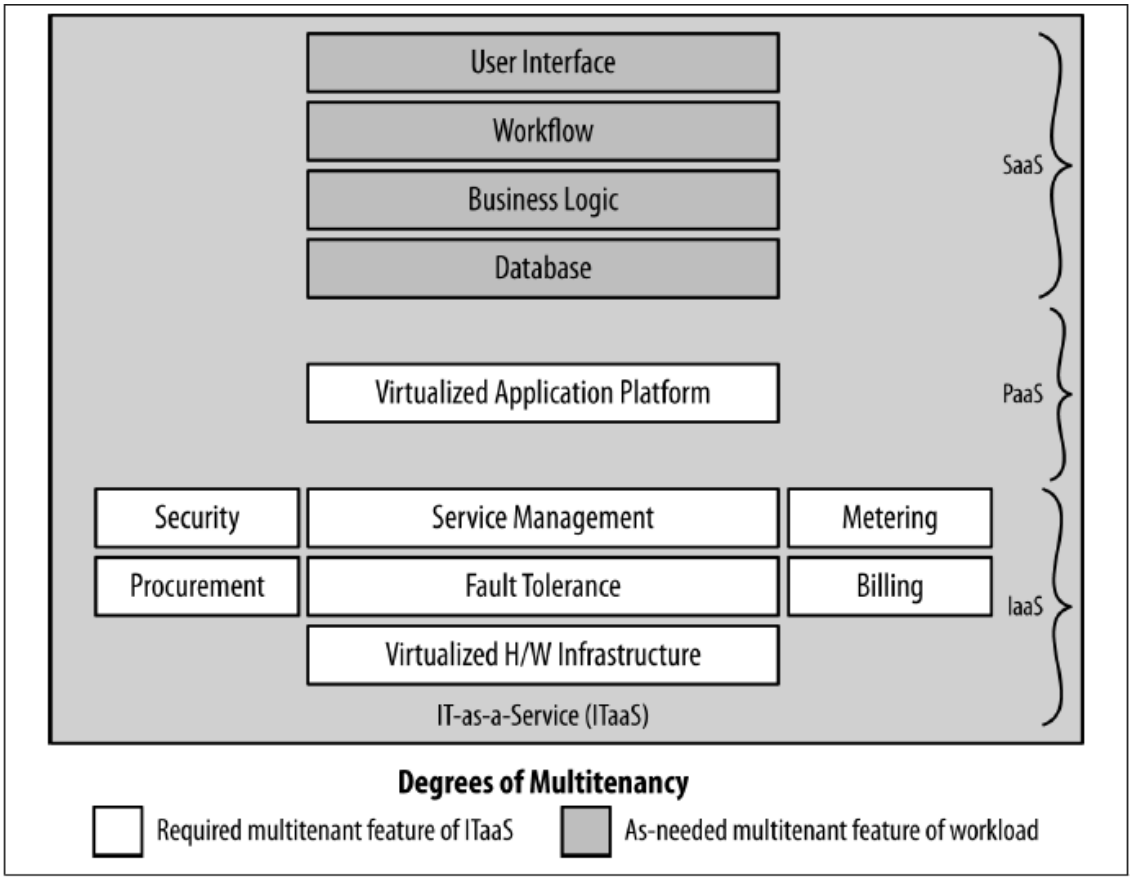

Figure 3.4: Degree of Multitenancy

-

various various instances of virtual servers which is called Virtual Machines, are run. So we need virtualized envrionment to serve the purpose.We also need traffic isolation because traffic from one tenant's user should not interact or should not interfere with the traffic with another tenant's user. VLANs are such networks.VLANs are in deployment since many years, but VLANs can't fulfill the needs of multitenant data centers.Because 12 bit VLAN id can only provide 4096 VLAN ids, which can be used to identify 4096 different machines. But in the multitenant data center environment, lots of virtual machines exist. So we need new standard or protocol or such mechanism which can handle these kind of requirements.

3.4.2 VxLAN protocol and packet flow

VXLAN can provide seamless layer2 connectivity over layer 3 networks.It can also isolate the traffic of one tenant's user from another tenant's user which is flowing in the data center. Basically it is a "MAC inside UDP" encapsulation. Basically VXLAN is layer 2

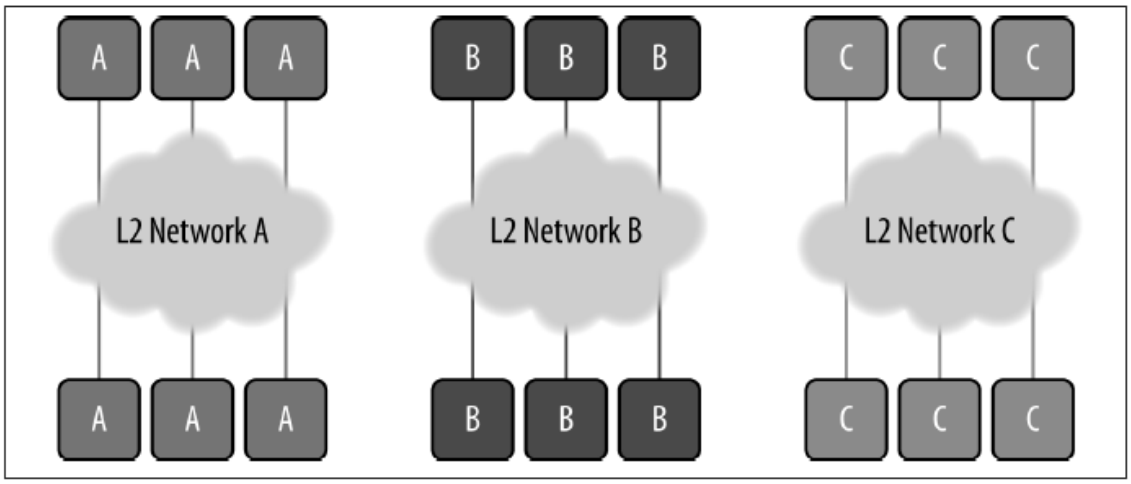

Figure 3.5: Logical view of network to each tenant

-

overlay protocol over the layer 3 network.Each overlay is called as VXLAN segment.VMs from different different segments cannot communicate with each other.Each segment is identified 24 bit identifier known as VXLAN network identfier(VNI).

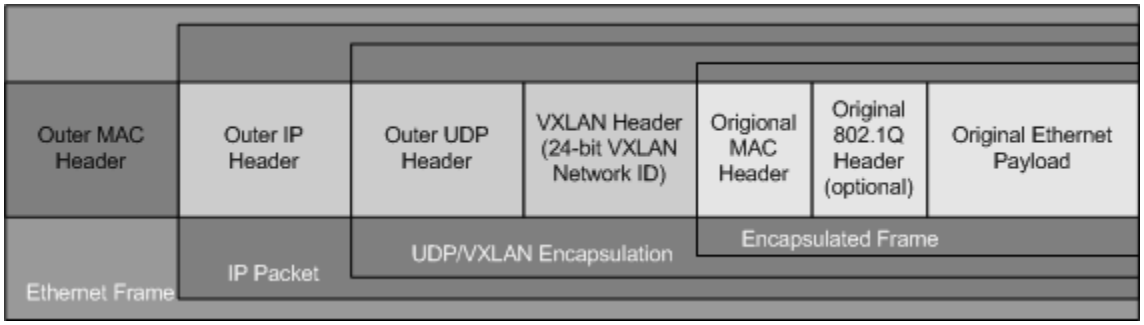

Figure 3.6: VxLAN Encapsulation

-

This 24 bit VNI allows 16 million VXLAN to coexist within the same administrative domain. Each VM is uniquely idenfied by MAC+VNI. So it is possible to have duplicate MAC address in the different VNI but not in the same VNI.

3.4.3 VXLAN packet flow

Let us assume that VM1 wants to send a unicast packet to VM2. VXLAN packet flow follows the steps described given below :

- VM1 needs MAC address of VM2, so that it can send the packet to VM2.

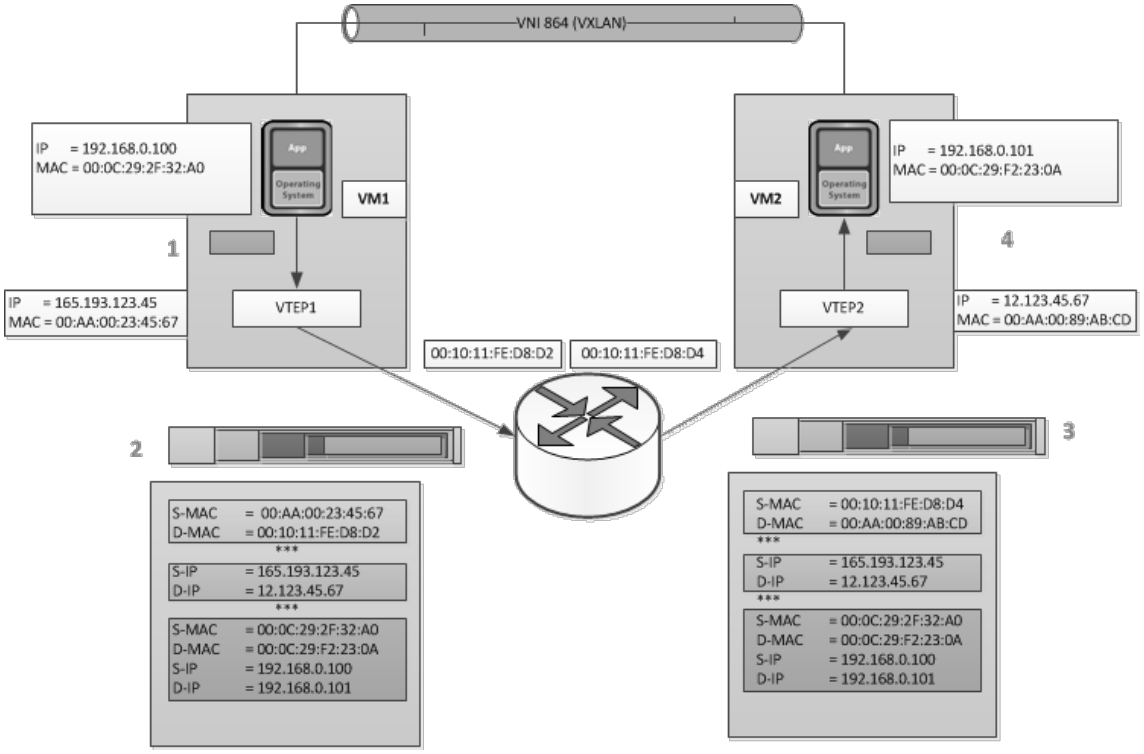

Figure 3.7: VxLAN Unicast Communication

-

- ARP request is sent by VM1 in order to find the MAC address of VM2.VM2's IP address is 192.168.0.101
- Tunnel endpoint VTEP1 will encapsulate the ARP request in a multicast packet to multicast group beloging to the VNI 864.
- All tunnel endpoints(VTEPs) sees that multicast packet and add the mapping of VTEP1 and VM1 to VXLAN tables.
- The packet is received at VTEP2 and it decapsulate the packet and send it to all the ports which are mapped with the VNI 864.
- VM2 gets the packet and sends ARP reply with its own MAC address.
- Now, VTEP2 will encpasulate the packet into unicast packet and send it back to the VTEP1 using IP routing mechanism.
- VTEP1 receives the packet, decapsulate the packet and sneds back to the VM1[9].
- Now that VM1 has MAC address of VM1.It will follow steps given below.

- VM1(192.168.0.100) will send the IP packet to VM2(192.168.0.101)
- VTEP1 will encapsulate the packet into VXLAN header as follows :
  * VXLAN header
  * Transport layer header which typically is UDP header
  * Network layer IP header
  * Layer 2header with the MAC address of the next hop
  * VTEP2 will receive the packet and decapsulate it. After decapsulating the packet, it will send the packet to VM2.
  * VM2 will receive the packet and wil treat it like any other IP packet.

Chapter 4

Implementation

This chapter elaborates the flow and various algorirthms used in the implementation of the validation of the layer 2 feature of the software model.

4.1 Implementation Flow

Complete packet with proper configuration is configured, and transmitted and expected packets are configured using scripting language.According to standard and specifications like RFCs, packets are configured.They should be in compliance with the standards.In the interface file, all the entry tables and registers are configured present in the chip. The output packets received should be in compliance with the expected packets. If there occurs an error, a report of bug is prepared to send it back to software implementation team.

4.2 Implementation of layer 2 features

Both L2 unicast and multicase features are validated in the layer 2 validation. Test cases are written in the TCL.I/o ports, ethertype,vlan tag and complete packet information is configured.In the interface file, all the entry tables are configured and all the required registers are enabled.Then corresponding input and output ports are enabled according to the test data.

Whenever a switch finds a match of VLAN id and MAC destination address in the entry table, packets are forwarded onto the output ports.In this process, switch learns the MAC source address.

L2 multicast is similar except sending a packet to a single port, it is sent to multiple

ports belonging to the same VLAN.Port Filtering Mode(PFM) is an important field, which controls the behavior of known and unknown multicast packet.For different PFM encodings, following actions are performed :

1. PFM value 0 : Both known and unknows multicast packets are flooded to VLAN members.
2. PFM value 1 : Unknown packets are floeeded to VLAN, and known packets are forwarded to receiving ports.
3. PFM value 2 : Known packets are forrwarded and unknown packets are dropped.

4.3 Algorithm for validating L2 unicast switching feature

1. Call functions or procedures to source all global files, configure global parameters and generate packets
2. Transmit packet configurations
   (a) Configure frame size, VLAN tag and transmitted ports.
   (b) configure MAC SA and MAC DA
   (c) Configure ethertype and frametye
   (d) configure payload L3
3. Expect packets configurations
   (a) Configured expected ports which are going to receive the packets, frame size and VLAN tag
   (b) Configure expected MAC DA and MAC SA
   (c) Configure expected frame type and ethertype
   (d) Configure expected OTAG,VID and outer priority
4. Configurations programmed in the interface
   (a) Initialize VLAN table for transmitting receiving ports

(b) Initialization of port table : configure VID(for priority and untagged packets only), spanning tree group id and learning mechanism of the switch.

(c) Configure CAM table to program hit for MAC DA and VID.

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

Figure 4.1: Layer 2 Test Scenarios

-

4.4 Algorithm for validating L2 multicast switching feature

1. Call functions or procedures to source all global files, configure global parameters and generate packets

2. Transmit packet configurations

   (a) Configure frame size, VLAN tag and transmitted ports.

   (b) configure MAC SA and MAC DA

   (c) Configure ethertype and frametye

   (d) configure payload L3

3. Expect packets configurations

   (a) Configured expected ports which are going to receive the packets, frame size and VLAN tag

   (b) Configure expected MAC DA and MAC SA

   (c) Configure expected frame type and ethertype

   (d) Configure expected OTAG,VID and outer priority

4. Configurations programmed in the interface

   (a) Initialize VLAN table for transmitting receiving ports

   (b) Initialization of port table : configure VID(for priority and untagged packets only), spanning tree group id and learning mechanism of the switch.

   (c) Configure CAM table to program hit for MAC DA and VID.

   (d) Configure multicast table with the group of the ports for multicast forwarding

4.5 Algorithm for validating Layer 3 IPv6 unicast routing feature

1. Call functions or procedures to source all global files, configure global parameters and generate packets

2. Transmit packet configurations

   (a) Configure frame size, VLAN tag and transmitted ports.

   (b) configure MAC SA and MAC DA

   (c) Configure ethertype and frametye as IPv6(0x86dd)

   (d) Configure Priority for the packets

   (e) Configure IP version 6, Destination Internet Protococol IPv6 address and Source IPv6 address

   (f) Configure IP version 6, Destination Internet Protococol IPv6 address and Source IPv6 address

3. Expect packets configurations

(a) Configured expected ports which are going to receive the packets, frame size and VLAN tag

(b) Configure expected MAC DA and MAC SA

(c) Configure expected frame type and ethertype

(d) Configure expected OTAG,VID and outer priority

(e) Configure IP version 6, Destination Internet Protococol IPv6 address and Source IPv6 address

(f) Configure IP version 6, Destination Internet Protococol IPv6 address and Source IPv6 address

4. Configurations programmed in the interface

(a) Initialize VLAN table for transmitting receiving ports

(b) Initialization of port table : Enable IPv6 routing coltrol for the port

(c) Configure Layer 3 forwarding table for destination hit, get next hop index as an output

(d) Configure Next Hop table pointed by next hop index to get the destination port number and destination VLAN id

(e) Configure interface table to change the outgoing MAC DA and MAC SA

4.6 Algorithm for validating Layer 3 IPv6 multicast routing feature

1. Call functions or procedures to source all global files, configure global parameters and generate packets

2. Transmit packet configurations

(a) Configure frame size, VLAN tag and transmitted ports.

(b) configure MAC SA and MAC DA

(c) Configure ethertype and frametye as IPv6(0x86dd)

(d) Configure Priority for the packets

(e) Configure source IPv6 address and multicast destination IPv6 address

(f) Configure IPv6 header

3. Expect packets configurations

   (a) Configured expected ports which are going to receive the packets, frame size and VLAN tag

   (b) Configure expected MAC DA and MAC SA

   (c) Configure expected frame type and ethertype

   (d) Configure expected OTAG,VID and outer priority

   (e) Configure IP version 6 header

4. Configurations programmed in the interface

   (a) Initialize VLAN table for transmitting receiving ports

   (b) Initialization of port table : Enable IPv6 routing coltrol for the port

   (c) Configure IPv6 multicast table, which gives the multicast index

   (d) Configure layer 3 replication table indexed by IP multicast index which provides destination port bitmap indicating the destination ports belonging to the multicast group address

   (e) Configure interface table to change the outgoing MAC DA and MAC SA

4.7 Algorithm for validating 6in4 Tunnel initialization

1. Call functions or procedures to source all global files, configure global parameters and generate packets

2. Transmit packet configurations

   (a) Configure frame size, VLAN tag and transmitted ports.

   (b) configure MAC SA and MAC DA

   (c) Configure ethertype and frametye as 6in4

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

Figure 4.2: Layer 3 Test Scenarios

-

(d) Configure Priority for the packets

(e) Configure IP version 6, Destination Internet Protococol IPv6 address and Source IPv6 address and complete IPv6 header

3. Expect packets configurations

(a) Configured expected ports which are going to receive the packets, frame size and VLAN tag

(b) Configure expected MAC DA and MAC SA

(c) Configure expected frame type and ethertype

(d) Configure expected OTAG,VID and outer priority

(e) Configure tunnel IPv4 header

(f) Configure inner IPv6 header

4. Configurations programmed in the interface

(a) Initialize VLAN table for transmitting receiving ports

(b) Initialization of port table : Enable IPv4 routing routing coltrol for the port

(c) Configure Layer 3 forwarding table for destination hit, get next hop index as an output

(d) Configure Next Hop table pointed by next hop index to get the destination port number and destination VLAN id

(e) Configure interface table to change the outgoing MAC DA and MAC SA and obtain the tunnel index

(f) Configure Tunnel encap table indexed by tunnel index and configure tunnel source IPv6 address and tunnel destination IPv6 address. This is important table for tunnel initialization/encapsulation

4.8 Algorithm for validating 6in4 Tunnel termination

1. Call functions or procedures to source all global files, configure global parameters and generate packets

2. Transmit packet configurations

   (a) Configure frame size, VLAN tag and transmitted ports.

   (b) configure MAC SA and MAC DA

   (c) Configure ethertype and frametye as 6in4

   (d) Configure Priority for the packets

   (e) Configure outer IPv4 header

   (f) Configure inner IPv4 header

3. Expect packets configurations

   (a) Configured expected ports which are going to receive the packets, frame size and VLAN tag

   (b) Configure expected MAC DA and MAC SA

   (c) Configure expected frame type and ethertype

   (d) Configure expected OTAG,VID and outer priority

(e) Configure only IPv6 header, because after tunnel termination, outer IPv4 header will be removed

4. Configurations programmed in the interface

   (a) Initialize VLAN table for transmitting receiving ports

   (b) Initialization of port table : Enable IPv6 routing routing coltrol for the port

   (c) Configure only IPv6 header, because after tunnel termination, outer IPv4 header will be removed

   (d) Configure Next Hop table pointed by next hop index to get the destination port number and destination VLAN id

   (e) Configure interface table to change the outgoing MAC DA and MAC SA and obtain the tunnel index

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)

Figure 4.3: Tunneling Test Scenarios

-

4.9 Results

- For layer 2 switching, 4 bytes of VLAN tag is added. Transmitted packet is the untagged packet, whereas received packet is the STAG 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 4.4: Layer 2 results

-

- For layer 3(IPv4) , MAC DA and MAC SA are changed. VLAN tag is added and TTL is decremented. Ethertype "0800" indicates that next header is IPv4 header.

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 4.5: Layer 3 switching results(IPv4)

-

- For layer 3(IPv4) , MAC DA and MAC SA are changed. VLAN tag is added and TTL is decremented. Ethertype "0800" indicates that next header is IPv4 header.

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 4.6: Layer 3 switching results(IPv6)

-

- For 4in6 tunnel encapsulation, inner IPv4 header is encapsulated inside outer IPv6 header. In transmitted packet, only IPv4 header is sent, but when received, the packet is seen as having the outer IPv6 header and inside that IPv4 header.

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 4.7: 4in6 tunnel encapsulation

-

- For 4in6 tunnel decapsulation, outer IPv6 header is removed and only customer header is seen in the packet, which is IPv4 header.

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 4.8: 4in6 tunnel decapsulation

-

Chapter 5

Future Work

5.1 Future Work

As VxLAN is one of the very important protocol in the data center environment, next activity is to have an algorithm to validate the VxLAN packet flows in the ESL validatino environment and have it tested for the switch. Also, Geneve(Generic Network Virtualization Encapsulation) draft is considered potentially a breakthrough protocol in terms of network virualization.Scope of Geneve protocol is to be studied too.

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, Oct 2010, pp. 61-62.

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

[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", 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.

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