A Thesis

entitled

Routing in VIMNet (A Virtual Modularized Internet Architecture)

by

Mohankrishna Vedam

Submitted to the Graduate Faculty as partial fulfillment of the requirements

for the Master of Science in Engineering

_______________________________________
Dr. Devinder Kaur, Committee Chair

_______________________________________
Dr. Henry F Ledgard, Committee Co-Chair

_______________________________________
Dr. Carvalho, Jackson, Committee Member

_______________________________________
Dr. Patricia R. Komuniecki, Dean
College of Graduate Studies

The University of Toledo

December 2010
An Abstract of

Routing in VIMNet (A Virtual Modularized Internet Architecture)

by

Mohankrishna Vedam

Submitted to the Graduate Faculty as partial fulfillment of the requirements for the
Master of Science in Engineering

The University of Toledo
December 2010

Present day Internet is as vulnerable as important and its applications are counting
from small e-mail applications to real time support between two people over Internet
through streaming the content sitting very far from each other. But the question comes
whether the Internet with its current TCP/IP static protocol architecture can handle it with
good Quality of service is the biggest challenge. There are many advances in this regard
to overhaul the entire network architecture and to replace the static protocol framework
with a dynamic one so that it makes the Internet more adaptable with the growth in
demand and also offer the best quality of service it can. Here in VIMNet which is a
Virtual Modularized Internet Architecture where the power of top layer’s functionalities
are led to that of end users such that there will not be a static protocol framework through
out the network. And this is achieved by providing a core set of services common to all
the participants in the network. Routing in VIMNet is implemented using Distance
Vector Routing the famous Link state routing method and the results found to be
substantial.
For my parents, fiancé, and friends
Acknowledgements

I wish to express my sincere thanks to my advisor Dr. Lawrence Miller and co-
advisor Dr. Devinder Kaur for their supervision, advice, guidance and support throughout
the work.

Special thanks to all my friends for their enthusiastic and generous support through
my research study.
Contents

Acknowledgements v

Contents vi

List of Tables viii

List of Figures ix

Introduction 1

1.1 Introduction .................................................................................................1

1.2 Target to be achieved ..................................................................................3

A Walk with Internet 5

2.0 Introduction ..................................................................................................5

2.1 ARPANET ....................................................................................................5

2.3 X.25 ..............................................................................................................8

2.4 ATM (Asynchronous Transfer Mode) .........................................................10

2.5 TCP/IP .........................................................................................................14

2.6 Drawbacks of current Internet architecture .............................................19

VIMNET 21

3.1 Introduction .................................................................................................21

3.2 VIMNET Architecture ...............................................................................21

3.3 Module management system ......................................................................24
List of Tables

2-1: Internet Protocol Layers. 14
4-1: Routing table 32
4-3: Routing Table in IGRP 34
List of Figures

1-1 Internet connections and its sample architecture 2
2-1: ARPANET sample architecture. 6
2-2: ARPANET communication. 7
2-3: X.25 message format 8
2-4: X.25 protocol suite in reference to OSI model 9
2-5: ATM packet 11
2-6: ATM Network with ATM switch and ATM endpoints. 12
2-7: ATM layers compared to OSI model. 13
2-8: IP packet with all the fields associated with it. 15
2-9: Transfer Control Protocol Packet with all the fields associated with it. 17
2-10: Network Protocols associated with Internet Protocol Suite 19
3-1: General network Structure 23
3-2: End Host system Model 24
3-3: Data travelling in VIMNet through routers that are equipped with dynamic protocols 26
4-1: Internet topology complexity for routing increasing day by day 29
4-2: Interior and Exterior Routing protocols 30
4-3: RIP hops count in a Network 32
4-4: Nodes in Interior Gateway routing protocol
4-5: Dijkstra’s algorithm shortest path first algorithm
4-6: BGP finite state machine
4-7: Basic route in VIMNet
4-8: VIMNet packet header
4-9: Flooding of the network to routers in the network with addresses.
5-1: The layout of routers arranged in Present Internet architecture
5-2: Average end-end delay in the layout as per the present Internet
5-3: Average end-end delay in the layout which is per VIMNet
5-4: Average jitter in the layout as per the present Internet
5-5: Average jitter in the layout as per the VIMNet
5-6: Last packet received in the layout as per the present Internet
5-7: Last packet received in the layout as per VIMNet
5-8: Throughput in the layout as per the present Internet
5-9: Throughput in the layout as per VIMNet
5-10: Total bytes received in the layout as per the present Internet
5-11: Total bytes received in the layout as per VIMNet
5-12: Total packet received in the layout as per the present Internet
5-13: Total packets received in the layout as per VIMNet
5-14: Bellman ford number of update in the layout as per the present Internet
5-15: Bellman ford no of updates sent in the layout as per VIMNet
Chapter 1

Introduction

1.1 Introduction

Internet is the basic means of communication in the modern world. It can also be described as the system that inter connects networks using the standard Internet Protocol Suite (TCP/IP) [16] helping all the users all over the world. It can also be described as a network of networks that supports multi user operated, public, private, and organizational and government operated networks which range from local to global scale and are interconnected through various cables and other means. There is nothing to be surprised if we say that world stops at once if Internet stops working. Now every walk of life includes and needs Internet.

Internet doesn’t have governance but have a set of rules to be followed to ensure service and those set of rules are called protocols. Internet runs on static protocol architecture. This static protocol is responsible for the connection, reliability, network management and flexibility and many more challenges to overcome in designing an Internet working architecture and it is no simple. There are many models which describe
how software in one system passes the data through the network to another system in another network such as OSI and TCP/IP [16] these are the standard reference models for today’s Internet.

Figure below depicts the Internet connections and its sample architecture originally from the English Wikipedia. Author Matt Britt This file is licensed under the Creative Commons Attribution 2.5 Generic license.

Figure 1-1 Internet connections and its sample architecture. (Wikipedia CC license free use for educational purposes obtained on Oct 31 2010)
Demand for Internet’s quality of service is rising in exponential way such that it’s no longer accepted if it delivers and breach in security or even delay in communication and transmission. End users must be ensured with reliability, scalability, robustness, adaptability of the architecture and that leads the development of many new and efficient routing protocols in the architecture of Internet even then we still have many issues that are needed to be addressed that comes along with the growth of demand and users for Internet. This leads us to a new architecture, an entire new way for communication and covering the flaws of existing architecture and it is VIMNet Virtual Modularized Network Architecture [20].

In VIMNet it provides a dynamic multidimensional protocol architecture which substitutes the static protocol architecture. It is designed in such a way that it will address and also accommodates the future networks which are mostly of very high speed wireless, multi-gigabit speed Ethernet and single as well as multi-wavelength fiber. Using the advantages of these extremely high speed mediums and the power of virtual circuit based technologies VIMNet is developed; here in VIMNet the concept of modularized protocols is used.

1.2 Target to be achieved

VIMNet architecture is equipped with dynamic modularized protocols which use virtual circuit /virtual path based system which we find in MPLS [7], ATM [1, 2, 7]. VIMNet consists of modules which are parts of the protocol which are all implemented with same Application Programmable Interface’s so that they can be used by the protocol framework. The basic protocol stack framework consists of protocol module manager
with basic set of protocols which are used to set up connection between the end host systems. Here we are implementing the basic Distance vector routing in the VIMNet module manager as the basic set of protocols for setting up connection. Thesis is organized as follows a walk with Internet will be held discussing the origin of Internet along with the protocols it came out, then listing out the challenges future Internet design should be included, Later in thesis description of VIMNet and its features. Routing in current Internet will be given a brief discussion along with the implementation of Distance Vector routing in VIMNet and simulation results along with the code.
Chapter 2

A Walk with Internet

2.0 Introduction

In this chapter we are going to discuss about how Internet came into existence what are the primary design specifications and goals of the Internet architecture. We will discuss about ARPANET, X.25, and TCP/IP and then we will discuss about the drawbacks and issues with the current architecture of Internet.

2.1 ARPANET

This architecture was named after the provider of funds to this project ARPA (Advanced Research Projects Agency) and it was designed in 1960. This architecture was developed for US Defense agency to develop a new packet-switching network technology. The backbone of ARPANET [21] consisted of packet switching nodes called IMP’s (Interface Message Processors) connected by 56Kbits/sec at that time.

In ARPANET as said the communication was formed by packet switching where the entire data is broke into small pieces called packets and these packets are used to
transfer from one node to another node. At its initial deployment of this ARPANET it was used to connect computers between four computers at University of California Los Angeles, Stanford Research Institute, University of California at Santa Barbara and the University of Utah. But when later it was expanded to few more computers ARPANET faced many compatibility issues and hence resulted in a new set of protocols called TCP (Transfer Control Protocol). The other disadvantage of ARPANET is that it failed in handling multiple network connections for different applications residing in the host computer which was later addressed in TCP with Network Control Program (NCP).

Figure 2-1: ARPANET sample architecture.
Communication between hosts with other hosts is done using messages. The groups of IMP’s (Interface Message Parser) make the hosts feel that they are talking to each other. The message which a host transfers is handed over to the source IMP (the IMP connected directly to the host). Then IMP routes each packet, which are pieces of message individually through the IMP-subnet, to the destination IMP. Packets are transferred along the subnet in such a way that each takes its own path which is estimated to be the shortest and it may vary with other packet. The destination IMP makes the destination feel that it is directly connected to the source node by re attaching the packets. It is more briefly explained in the following diagram.

Figure 2-2: ARPANET communication.
2.3 X.25

X.25 [4, 21] is one of the routing protocols, which were used before OSI model came into existence. The only prominent routing protocol that was in use before X.25 is 1822(ARPANET). X.25 was used purely for packet switching networks across computers. It was developed at ITU-T (International Telecommunication Union-Telecommunication Standardization Sector) [5]. It got three layers which is similar to the lower three layers of TCP Physical, Data link and packet layers. It is made to support connection-oriented networks (telecommunications). ARPANET was mainly a project of DoD where fault tolerance was the only concern so the packets are routed in such a way that lose of packets result in rerouting and doesn’t affect reaching the destination. But when it comes to X.25 the primary goal is maintaining the quality of service as it deals with telecommunications. A computer setting up a connection in X.25 mode shall first send a request for opening a connection with a remote computer by sending a connection setup request packet, and this connection when accepted shall be given a connection number. And this connection is used to transfer data packets the sample X.25 message packet consists of a 3 byte header and up to 128 bytes of data, the header consists of a 12 bit connection number a packet sequence number, an acknowledge number and a few miscellaneous bits.

![Figure 2-3: X.25 message format](image_url)
The Communication mode in X.25 introduces a new concept called Virtual Circuit which is a logical view of a connection which feels as if it is directly connected to the remote computer but in physical terms it is passed through many number of Packet Switching Equipments. And this many such virtual circuits comprises of the X.25 communication method, This X.25 virtual circuit is setup with remote computer as soon as the connection is established and this circuit is terminated when the packet transfer is done between the two Data Terminal Equipments. Physical circuit may contain many Virtual Circuits (logical one). The following diagram depicts the protocol suite of X.25.

![Figure 2-4: X.25 protocol suite in reference to OSI model (courtesy: CISCO systems)](image)

The X.25 protocol suite consists of three layers PLP (Packet Layer Protocol), LAPB (Link Access Procedure, Balanced), and X.21 layer.

Packet Layer Protocol is the network protocol of X.25, which deals with the connection management between Data terminal equipment and the packet switching equipment. This layer mainly operates in five distinct modes call setup, data transfer, idle, call clearing and restarting.
Link Access Procedure, Balanced is the Data-Link layer of X.25 and this layer is responsible for the communication between PLP and X.21 layer in other words the main channel for communication between DCE (Data Circuit Terminating Equipment) and DTE this ensures the packet transfer and error handling and ordering of packets.

X.21 is the physical layer of X.25 and it handles the hardware for connecting and disconnecting with the circuits. It has the physical control over the entire communication it is responsible for the connection between the physical layer medium and the packets and its form of communication.

Apart from its advantages of being a virtual circuit concept implementer it faces very long and variable end-end delay and also low throughput and this technology was a mixed success and Frame Relay successor to frame relay overcame some of its disadvantages by having no error and flow control.

2.4 ATM (Asynchronous Transfer Mode)

ATM (Asynchronous Transfer Mode) [2, 3] is another important connection-oriented protocol where it doesn’t use the method adopted by telephone system or X.25, which is synchronized to a clock where ATM is not. In ATM sending data starts with a connection setup between two systems (far from each other). As the connection setup packet traverses all the nodes in the subnet each node keeps the entry of the node and calculate its resources for communications in its internal tables. These connections are often called as virtual circuits; ATM circuits are mostly virtual circuits and can also be permanent circuits.
Permanent Virtual circuits are often used in ATM networks are similar to leased telephone lines where the connection is temporary and has a unique connection identifier. The entire idea between ATM networks is the usage of cells, which are small fixed size data packets. They are 53 bytes long of which 5 bytes are header and the remaining 48 bytes are the payload.

![Figure 2-5: ATM packet](image)

Header contains part of data which checks the connection identifier which makes the intermediate hosts to identify who is the sender and where the cell belongs to, the main advantage in ATM is that the hardware associated to it does the cell routing at a very high speed. This is because of the fixed size cells which are easy for the hardware routers to be built in a manner that it handles short, fixed length cells where as variable size packets are handled by some software which makes the data transfer slower. ATM network comprises of ATM switch and ATM end points. ATM switch holds the responsibility of the cell transmission and the ATM end points are those who send or receive the data formatted into cells. ATM can be operated in two different cell formats, NNI (Network-Network Interface) and UNI (User-Network Interface)
ATM protocol Layer has three layers Physical Layer, ATM Layer and ATM adaption layer apart from these three layers end users have the advantage of adding extra layers on the top of these layers.

Physical Layer:

This layer deals with Physical medium such as Voltages, and bit transmission and bit timing along with many other issues, ATM that is designed in a mode that makes it independent of transmission medium

ATM Layer:

This layer deals with cells and transportation of cells. This layer is responsible for all actions of virtual circuits including connection setup and connection termination of VC’s and also handles congestion in Virtual Circuit’s.
ATM Adaption Layer:

This layer is responsible for dissembling and assembling of large chunks of data into small cells and thus this layer is responsible for the faster performance of ATM. [1] this layer offers

1. CBR - Constant bit rate: a Peak Cell Rate (PCR) is specified, which is constant.
2. VBR - Variable bit rate: an average cell rate is specified, which can peak at a certain level for a maximum interval before being problematic.
3. ABR - Available bit rate: a minimum guaranteed rate is specified.
4. UBR - Unspecified bit rate: traffic is allocated to all remaining transmission capacity.

And this layer follows traffic shaping which is responsible for the traffic coming into ATM network and also going away from ATM. The following is the pictorial representation of ATM layers. The services of the ATM layers are divided as follows AAL-0 NULL AAL, AAL-1 Supports Class 'A' Constant Bit Rate Traffic, AAL-2 Variable Rate Synchronous Service, AAL-3/4 End-To-End Data Transport, AAL-5 Simple Data Transfer.

Figure 2-7: ATM layers compared to OSI model. (Courtesy from Wikipedia creative commons license)
2.5 TCP/IP

TCP/IP [16] is a set of protocols built by DARPA that evolved from ARPANET in 1970 and it is used for communications in Local Area Networks (LAN) Wide Area Networks (WAN). TCP/IP provides end-to-end connectivity with specific guidelines for transferring, formatting and routing of packets through networks with robustness and scalability. Later there emerged many algorithms and protocols that had an efficient way of transmission in the network. TCP/IP, which stands for Transmission Control Protocol and Internet Protocol which together commonly, called as Internet Protocol Suite. The TCP/IP (RFC 1122) similar to other protocols and it contains layers in its architectural model that describes how and when information is transmitted along the Internet.

<table>
<thead>
<tr>
<th>Application Layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transport Layer</td>
</tr>
<tr>
<td>Internet layer</td>
</tr>
<tr>
<td>Network access Layer</td>
</tr>
</tbody>
</table>

Table 2-1: Internet Protocol Layers.

The main protocols of Internet Protocol suite are TCP and IP i.e., Transfer Control Protocol and Internet Protocol.

Internet protocol:

This is one of the protocols that deals with routing in the networks this deals with addressing and identification of the network host and also packets routing. Here all the
choice for packet switching network on a connection less network is made. Along with TCP, IP [21, 22] is considered as the heart of the Internet protocol suite. IP networks are still sub divided into subnets. Subnetting makes the administration more effective with several other benefits such as better usage of the network and its network addressing. One such benefit includes that the network in the outside world considers that the organization has only one network and this subnet deals with the internal network hierarchy. The Internet protocol packet is of 32 bit size which is still divided into 4 bit version and 8 bit header and 16 bit type of service and many other fields the following picture depicts the one such packet with all the fields.

![Figure 2-8: IP packet with all the fields associated with it.](image)

Version—indicates the version of IP currently used.

- IP Header Length (IHL)—indicates the datagram header length in 32-bit words.
• Type-of-Service—Specifies how an upper-layer protocol would like a current datagram to be handled, and assigns datagrams various levels of importance.

• Total Length—specifies the length, in bytes, of the entire IP packet, including the data and header.

• Identification—contains an integer that identifies the current datagram. This field is used to help piece together datagram fragments.

• Flags—consist of a 3-bit field of which the two low-order (least-significant) bits control fragmentation. The low-order bit specifies whether the packet can be fragmented. The middle bit specifies whether the packet is the last fragment in a series of fragmented packets. The third or high-order bit is not used.

• Fragment Offset—indicates the position of the fragment's data relative to the beginning of the data in the original datagram, which allows the destination IP process to properly reconstruct the original datagram.

• Time-to-Live—maintains a counter that gradually decrements down to zero, at which point the datagram is discarded. This keeps packets from looping endlessly.

• Protocol—Indicates which upper-layer protocol receives incoming packets after IP processing is complete.

• Header Checksum—helps ensure IP header integrity.

• Source Address—specifies the sending node.

• Destination Address—specifies the receiving node.

• Options—Allows IP to support various options, such as security.

• Data—Contains upper-layer information.
TCP: Transfer Control Protocol

This protocol is responsible for the reliable packet transformation. TCP provides many features such as stream data transfer, reliability, efficient flow control, full-duplex operation, and multiplexing. This is considered as a reliable connection oriented protocol that allows a byte stream originating on one machine to be transferred to other machine without any error in the Internet. Another feature of TCP is sliding window protocol that conceptually considers each portion of the transmission as a unique consecutive sequence with a sequence number. And this helps the end receiver to identify and place the received packets in correct order through discarding duplicate packets and identifying missing one’s and request for retransmission. The disadvantage of this protocol is that there is no limit in size of the sequence number it carries; by placing limits on the number of packets sliding window protocol carries unlimited number of packets to be transmitted using fixed size sequence numbers.

![Figure 2-9: Transfer Control Protocol Packet with all the fields associated with it.](image)
Source Port and Destination Port—Identifies points at which upper-layer source and destination processes receive TCP services.

- Sequence Number—usually specifies the number assigned to the first byte of data in the current message. In the connection-establishment phase, this field also can be used to identify an initial sequence number to be used in an upcoming transmission.
- Acknowledgment Number—contains the sequence number of the next byte of data the sender of the packet expects to receive.
- Data Offset—Indicates the number of 32-bit words in the TCP header.
- Reserved—Remains reserved for future use.
- Flags—carry a variety of control information, including the SYN and ACK bits used for connection establishment, and the FIN bit used for connection termination.
- Window—Specifies the size of the sender's receive window (that is, the buffer space available for incoming data).
- Checksum—indicates whether the header was damaged in transit.
- Urgent Pointer—Points to the first urgent data byte in the packet.
- Options—Specifies various TCP options.
- Data—Contains upper-layer information.

The Internet protocol apart from having Internet protocol at Internet layer and transfer control protocol at transport layer also offers many services at application layer such as file transfer (ftp), terminal (telnet), e-mail (smtp) and supports many other applications built upon them.
2.6 **Drawbacks of current Internet architecture**

With the advent of technology that changed the face of world had also changed the way Internet is and it also demanded the features of future Internet that was made into real with all possibilities up to some extent. The growing usage of Internet has been the first and foremost challenge to Internet architecture which was designed many years ago and now with the number of end users of Internet it has been a nightmare to accommodate the number with the addressing which the Internet is following now and even with the usage of IPv6 the intensity was been minimized but not solved.

Firstly with too many protocols for transferring data from one end to other the issue to be foreseen is the complexity of the Internet architecture and the cost for end-to-end transmission. There are no particular standard protocol for data transmission and even this increased burden on both user and the medium perspective.

Environment is also should be considered when discussing about the challenges to the Internet as now the Internet is trying to accommodate various environment such as windows, Unix which increases the cost of transmission in decoding and encoding the data at the physical layer and application layer levels.
Security remains the most challenging issue with all types of network architecture this stands as first and foremost issue that should be addressed while designing architecture of any network even at granular level as any sort of security vulnerabilities cannot be tolerated with the usage of Internet the misuse comes along the way. Addressing the security issues of the present Internet like packet filtering controlling virus, Trojans, malware advent. Networking Internet is an issue that will be static throughout operation of Internet, as network architecture will be fixed during the setup. Any delays, service breakdowns cannot be accepted. When a packet is transmitted to a destination it is expected with a minimal delay and reach the destination, if at all network failure occurs that shouldn’t affect the transmission with a greater impact.

There are many other challenges inclusive to the above-specified ones which future Internet is expected to address.
Chapter 3

VIMNet

3.1 Introduction

Dynamic Virtual Modularized Network Architecture is one of network architectures that answer most of the issues and challenges with the current Internet architecture, keeping in mind the three unalterable factors. First is that the Internet will always be a network of networks. Second, the bandwidth limitation of wired connections will be bounded by the theoretical limitations of fiber. Finally, wireless communication will be bounded by the restrictions of Maxwell’s equations. Thus the design of the future Internet is made keeping these constraints in mind and also providing ability to come over the weakness of these factors.

3.2 VIMNet Architecture

The architecture is entirely different from the existing protocol architecture where the Internet is entirely dependent on the core routers that are extremely powerful packet switching routers interconnected by very high-speed fiber connections. On top of them
another hierarchy of network exists that includes Ethernet and wireless at higher speeds and all these are interconnected, which is Internet. In this kind of network once when the deployment begins the request for new services and features increase that are poorly supported or not supported at all by TCP/IP. There are many new virtualization methods that are already implemented which are running with lower success in the present packet switching networks.

VIMNet architecture starts with changes to the present architecture and utilizing the aid for laying fiber optical fibers in the ground. The core routers in the Internet are replaced by connection oriented routers in 5X-10X in number and this trying to eliminate usage of connection less routing which tries to provide reliable connection with heavy weighted protocols. In such scenarios the congestion increases along with loss of network bandwidth as these results in retransmission of packets through this complex architecture. In VIMNet these core routers helps in providing core framework where the routers are interconnected in mesh network manner with single and multi high-speed fiber optical trunk connections. Each trunk connection will contain redundant primary data path groups and private path groups. For each primary data path group, there will be at least one private connection that will be used for a variety of uses including security, router management, and connection setup and connection management. On top of these core router services comes another series of hierarchal networks that accommodate the use of high-speed wireless and multi gigabit Ethernets for transmission. The following picture depicts the base deployment of VIMNet.
Figure 3-1: General network Structure (courtesy article from VIMNet by L Miller
IEEEDOI: 10.1109/EIT.2007.4374521)

The hardware that associated with this architecture is similar to the routers used in
ATM but instead of using fixed packet size it does handle variable sized packets, which is
based upon the constraints and capabilities of the connection. These network cards
provide the flexibility for processing of data that is extended from core of router to the
network card such that many custom built technologies such as encryptions can be
implemented in short these network cards behave like a mini computer which include
CPU that can gain the additional resource provided by core services of VIMNet.
3.3 Module management system

The smallest defined object of the model VIMNet is considered as module, which is similar to layer in Internet architecture, but restricted only to visualization. Module’s primary responsibility includes communication with modules located in different protocol stacks that handle all the communication responsibilities associated with connection and handling it. In VIMNet protocol module doesn’t have any specific role except for that it should support the core service module which provide dynamic protocol framework which makes it easier to deploy algorithms that does the job which ranges from small mathematical calculations to highly complex p2p mesh network architecture. The module management system replicates the protocol stack that is dynamic in the case of VIMNet, each virtual protocol stack that makes a router or an application program to feel like single network connection and it is generated at the time of connection setup.

Figure 3-2: End Host system Model (courtesy article from VIMNet by L Miller IEEEDOI: 10.1109/EIT.2007.4374521)
The virtual protocol stack that is built consists of connection establishment packet and contains the end host connection data, list of modules the end host support and a connection mapping for the router modules (routing path) and any additional data if needed. As the packet passes through the network building a virtual circuit the requested router modules are loaded and connected and after reaching the destination it build upon the protocol stack and will send an acknowledgement so that the actual data can be transmitted this enhances the data transfer as VIMNet architectural requirement is provisioning of Redundant data paths (RDP) which in turn will be split into Primary Data Paths (PDP) and Redundant Primary Paths (RPP). The ideally it is expected to have several primary data connections; this path enables the module on a local router to de-multiplex and redirect packets to correct path.

The flexibility of having dynamic protocol architecture for VIMNet with Module Management system taking care of the protocols built upon enhances the features of data transmission in many ways as in developing an encryption technique that is dedicated between two end hosts that doesn’t have any impact on any other connections provided through the router.

The data which started at the higher layer web browser at the end-host 1 which looks similar to a protocol suite of any other regular Internet architecture but it differs in the number as it is not fixed at any time it varies with the architecture and it can be any in number this data passes through the router and reaches the end host2 with prior intimation for the protocol stack to decrypt the data which was encrypted at end-host1.
3.4 Advantages of VIMNet over present Internet Architecture:

VIMNet is a dynamic protocol architecture built keeping in mind the challenges that are being faced by present day Internet and having added the Virtual circuit methods that proved success in data transfer and with slight changes to it making VIMNet a powerful architecture.

Security: In VIMNet data transfer is done using redundant primary paths and primary data paths so the data is split into these through so that data is so secure. Whereas in present day Internet it is not secure enough as packets can be hacked and information can be taken away.

Complexity: It is less complex compared to present day Internet as it does have one fixed layer in its architecture and is open for building still more layers on it, whereas Internet has fixed number of layers which got predefined responsibilities.
Congestion: It is very low in VIMNet as because of the customized routers and fixed path for various connections. Where as in current Internet the congestion is so high that the routers are not capable of handling traffic and packets may follow different connections.

Reliability: VIMNet has the good terms of reliability as it is virtual circuit switching and also connection oriented service so the probability of packet loss is so less where as current Internet is connection less and packet switching where the loss of packets is more.

Packet size: It is very small in Internet which can hold less information and more processing time where as VIMNet has large packet size and got very little processing time.

Bandwidth Utilization: this is another important factor that should be counted for where it is not satisfactory in case of Internet as it uses small packets and it is not in the case of VIMNet, which has excellent network bandwidth utilization as it allows large packets for data transfer.

Apart from above quoted advantages VIMNet has the feature of adding extra modules depending upon the end user choice. The network’s infrastructure and dynamic protocol framework provides the desired flexibility for the needs of future applications, while also providing facilities for increased QoS, reliability, real-time support and other application based constraints.
Chapter 4

Comparison of routing in Internet and VIMNet

4.1 Routing in Internet

Routing is the method of selecting route to communicate between two networked end systems. Route is also called as path that is between two routers, which are connected either in connection oriented or connection less mode. Routing in another sense means connection between structured network addresses as those structured address can be made an entry into routing table trough which routing decisions are made. Routing can be broadly classified as Unicast, Broadcast, and Multicast, this classification is made on the basis of the operation of network where Unicast routing mode delivers message to a single node, in broadcast routing delivery is made to all routers in networks, where as multicast is the mode where messages are sent to group of nodes that have expressed interest in receiving the message.
4.2 Basic routing protocols

Routing protocols are the protocols, which specify how routers should communicate with each other in how they should choose route in the network and also exchange the knowledge of the networks they are in prior to connection setup. Routing protocols can be divided into two main categories based on the mode and place they operate they are Interior routing protocols, Exterior Routing Protocols [21, 22]. The main responsibilities of a routing protocols are they should be designed keeping in mind the way to prevent loops from forming or if happens breaking them down, it is also important to select a route calculating the hop cost on entire route as it is not advisable to adopt a route has highest hop cost which in turn effects QoS, and also scalability of route.

Figure 4-1: Internet topology complexity for routing increasing day by day. (Connections of all the sub-networks in the world) Credit: Bill Cheswick, Lumeta Corp
4.2.1 Interior routing protocols

Interior routing protocols or interior gateway protocols are protocols that are used for communication within an autonomous system (AS). An Autonomous system is a collection of systems (routers) that can be grouped into one as they share same or more network operators that possess same policy for routing; they are identified with a numbering called Autonomous System Number (ASN). These interior routing protocols work within or for AS. The different routing protocols are Distance Vector Routing Protocol and Link State Routing Protocol. To generalize these routing protocols are more for LAN’s but not limited to.

Figure 4-2: Interior and Exterior Routing protocols (courtesy CISCO learning press)
4.2.1.1 Distance vector routing

Distance Vector Routing uses Bellman-Ford algorithm where each router doesn’t have information about the entire network, it gathers information by advertising the information it has regarding its first level of neighbor nodes (here routers) to other nodes and receiving such advertisement which contains the same details so that it starts updating its routing table upon that. This process continues until all the routes to respective routers are updated in routing table to stable values.

The protocols that are based on Distance Vector Routing are

1. Routing Information Protocol
2. Interior Gateway Routing Protocol
3. Enhanced Interior Gateway Routing Protocol

Routing Information Protocol:

This routing protocol is based on Open shortest Path First (OSPF) [15, 17, and 18]; protocol is built upon hop count as a routing metric. RIP terms the unit of moving from one node to other as a hop and the time limit is 180 seconds, RIP avoids formation of loops by implementing limit on number of hops it allows in a route from a node to other node. The maximum number of hops that should happen in RIP is limited to 15 that also state the scalability limit on the number of nodes a network should possess.
Table 4-1: Routing table

<table>
<thead>
<tr>
<th>Node/via</th>
<th>Two</th>
<th>Route</th>
</tr>
</thead>
<tbody>
<tr>
<td>One</td>
<td>0</td>
<td>One</td>
</tr>
<tr>
<td>Two</td>
<td>1</td>
<td>One-two</td>
</tr>
<tr>
<td>Three</td>
<td>2</td>
<td>One-two-three</td>
</tr>
</tbody>
</table>

The above is the routing table for the network. The limitation of RIP is if the hop count is more than 15 ex: if the hop count is 16 the distance to reach is counted as infinity and the node is declared as not reachable. RIP works for small networks with good routing efficiency.
**Interior Gateway Routing Protocol:**

This protocol was invented by Cisco to overcome RIP issues such as limitation to 15 hops and only considering hop count as the sole metric in designing route. IGRP was built keeping in mind that the route should be stable in very large or complex networks which should never possess any loops, it should be adaptive to network changes, and it shouldn’t consume more bandwidth (lower overhead), uniformly distributing traffic. IGRP is designed such that it can handle different types of protocols.

![Diagram of nodes in Interior Gateway routing protocol](image-url)

**Figure 4-4: Nodes in Interior Gateway routing protocol**

The role of IGRP comes into existence making the routers to easily identify the nearest gateway to another network’s router for setting up communication. In the following diagram where a gateway is connected through two Ethernets 128.6.4.0 and 128.6.5.0 and a subnet at 10.0.0.0 that is trying to reach 128.6.21.0 should identify that 128.6.5.0 is its nearest gateway.

The routing table will contain the gateway and interface details for each router the packets that are addressed for the networks 128.6.5 and 128.6.4 will be simply directed using Ethernet interface but and the router at 10.0.0.0 finds the router 128.6.5.0 as the nearest router for communication by the indication provided by routing table about which gateway should be used for communication. The gateway at the time of its
 initialization will be loaded with details about the network such as topological delay, bandwidth of the link. And all other gateways in the network are programmed such a way that they exchange the information about the gateways and details about the network with the corresponding gateway, in this manner every packet for communication will reach every gateway in the system provided the network is fully connected.

Table 4-2: Routing Table in IGRP

<table>
<thead>
<tr>
<th>NETWORK</th>
<th>GATEWAY</th>
<th>INTERFACE</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.6.4</td>
<td>none</td>
<td>Ethernet 0</td>
</tr>
<tr>
<td>128.6.5</td>
<td>none</td>
<td>Ethernet 1</td>
</tr>
<tr>
<td>128.6.21</td>
<td>128.6.4.1</td>
<td>Ethernet 0</td>
</tr>
<tr>
<td>128.121</td>
<td>128.6.5.4</td>
<td>Ethernet 1</td>
</tr>
<tr>
<td>10</td>
<td>128.6.5.4</td>
<td>Ethernet 1</td>
</tr>
</tbody>
</table>

**Enhanced Interior Gateway Routing Protocol:**

Enhanced Interior Gateway Routing Protocol (EIGRP) is also a interior gateway protocol invented by CISCO, which is a much advanced distance vector routing protocol which can minimize the routing instability after topology changes happen an also the optimizes bandwidth utilization. The data collected by EIGRP is stored as three tables namely Neighbor table, Topology table, Routing table where

Neighbor table stores data about neighboring routers that are accessible through directly connected routers.
Topology table contains data that is an aggregation of routing tables that is gathered from all directly connected neighbors. This table contains details of network with respective metrics such as for every destination its successor and feasible successor, along with every route marked as active or passive depending on the topology changes.

Routing table contains data about routes to all destinations. The routing table is populated from the topology table with every destination network that has its successor and feasible successor (if exists).

4.2.1.2 Link state routing

In Link state routing each node will possess details about entire network topology, resulting in calculation of best route to destination by each node independently. In DVR protocol nodes share information about entire topology information at node level. The different routing protocol that uses link state routing method is Open Shortest Path First (OSPF)

Each router in Link state Routing should do the following

1. Discover its neighbors and learn their networking addresses.
2. Measure the delay or cost to each of its neighbors
3. Construct a packet telling all it has just learned
4. Send this packet to all other routers
5. Compute the shortest path to every other router
Open Shortest Path First (OSPF):

It is an Adaptive Routing protocol for Internet Protocol networks. It uses Link state routing protocol and operating within a single Autonomous system (AS). OSPF routes packets solely within a single AS by gathering link state information from available routers and construct a topology map of the network. It calculates the shortest path tree for each route using a method based on Dijkstra’s algorithm, which is a shortest path first algorithm.

<table>
<thead>
<tr>
<th>Vertex</th>
<th>Known</th>
<th>Cost</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>true</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>1</td>
<td>true</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>true</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>true</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>true</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>true</td>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>

Figure 4-5: Dijkstra’s algorithm shortest path first algorithm

In the above illustration routing to the entire network happens with a fundamental data map which is graph of the network to create this the entire network is flooded with information about other nodes it can connect to and each node then independently determines the least cost path from itself to every other node using a standard shortest path. The result is a tree-rooted t the current node such that every route from a node on the tree can be reached at lowest cost. This tree is used to construct routing table. The algorithm for finding shortest path for each node is as follows
function Dijkstra(Graph, source):
    for each vertex v in Graph: // Initializations
        dist[v] := infinity ; // Unknown distance function from source to v
        previous[v] := undefined ; // Previous node in optimal path from source
    end for ;
    dist[source] := 0 ; // Distance from source to source
    Q := the set of all nodes in Graph ; // All nodes in the graph are unoptimized - thus are in Q
    while Q is not empty: // The main loop
        u := vertex in Q with smallest dist[] ;
        if dist[u] = infinity:
            fi ;
        remove u from Q ;
        for each neighbor v of u: // where v has not yet been removed from Q.
            alt := dist[u] + dist_between(u, v) ;
            if alt < dist[v]: // Relax (u,v,a)
                dist[v] := alt ;
                previous[v] := u ; fi ;
        end for ; end while ;
    return dist[] ;
end Dijkstra.
OSPF is the first deployed popular routing protocol that can guarantee loop-free paths, OSPF works well and good even with a network, which has, more than 1000 routers provided the network must be designed to minimize overhead to achieve stable operation.

4.2.2 Exterior routing protocols

Exterior routing protocols deals with routing between Autonomous System unlike within Autonomous System’s as in case of interior routing protocols. The famous Exterior routing protocol is border gateway protocol (BGP) [13, 14]

4.2.2.1 Border gateway protocol

The role of Border Gateway Protocol comes in use when the communication is between Autonomous System’s, as a router in one Autonomous System if needed to transmit data to another router in another Autonomous System will not care about transit packets (used to regulate traffic in Autonomous System) as it will be an extra overhead if needed it will send any transit packets upon request by other Autonomous System. In short Internet service provider will be very much happy to transit packets to its consumers but not other’s BGP allows such kind of routing policies in inter Autonomous System communications.

BGP routing constraints will be as no transit traffic through certain Autonomous System’s and such policies are embedded into the BGP router but not in the protocol.
BGP in order to make decisions in operations with other peers uses the above depicted Finite State Machine which contains six states Idle; Connect; Active; Open Sent; Open Confirm; and Established. For each peer-to-peer session, a BGP implementation maintains a state variable that tracks which of these six states the session is in. The BGP protocol defines the messages that each peer should exchange in order to change the session from one state to another.

### 4.3 Routing in VIMNet

Routing in VIMNet is different from its counter parts as VIMNet is a blend of network hardware and also with Virtual Circuits, where virtual circuits give the utmost Quality of service with their method of routing in which a virtual circuit is formed and dedicated to the communication channel. This dedicated communication channel creates an uninterrupted data flow. VIMNet data packet size is similar to that of ATM but with an extra padding of 4 bytes to the predefined size of packet of 52 bytes that stores store
the virtual circuit flow order (as per S.Bolla in Implementation of Virtual Circuits in VIMNet).

![Figure 4-7: Basic route in VIMNet (courtesy article from VIMNet by L Miller IEEEDOI: 10.1109/EIT.2007.4374521)](image)

The Implementation of Virtual circuit’s helps in attaining high Quality of service as virtual circuits eases the complexity for a network from maintenance perspective as packets that flow through virtual circuits which are in a dedicated connection will follow the same path to the destination. This will help the network in a lot of ways. Apart from reducing the risk of organizing traffic at the time of creation of the route during connection establishment time, the connection is released once the data reaches its destination, thus, releasing the routes for further use. It also reduces the overhead required to maintain reliability, breaking up of data into packets and finding all the possible routes to reach the destination. Routing in VIMNet makes the effective bandwidth issue as in today’s network architecture that is more complex and also tries to give better Quality of service while losing the control over effective bandwidth utilization. As this QoS terms make the bandwidth broker or Internet Service provider to
understand the service policies that understands the user’s protocol so for better QoS ISP allocates a static provision of resources those results in Bandwidth gap, which is wasted.

![VIMNet packet header](image)

Figure 4-8: VIMNet packet header

The above is proposed VIMNet packet header that contains the VCI (Virtual Circuit Identifier) CLP (Cell Loss Priority) H (Number of Hops made), PT (Payload Type), HEC (Header error Checksum) the implementation of H bit adds the Virtual Circuit formation a extra information about the connection status how many hops it covered so far and at each subnet router this bit is reset and it’s counter starts from there adding each hop it made to reach the destination. Subnets will maintain a table (as in Link state routing) and generates the shortest path to reach the destination and then the entire network will be having a overview of the network graph which is used in making the routing decision thus adding the effectiveness of Link state routing to VIMNet enables the understanding of network and rerouting the packets are denying the connection establishments (if any) such that bandwidth and congestion issues will be solved.

Routing in VIMNet is handled in a way that it overcomes the issues of Link state routing as in VIMNet it is not building up the overhead for the entire network on packets which is not in the case of Link state routing and utilizing the V.C’s makes it bandwidth effective. The entire routing is divided into four steps.

Step1: Flooding the entire network with initial network address

Step2: Packet is initiated at sender with destination router address
Step3: Each router at the end of Virtual Circuits updating the network graph

Step4: Error handling and routing

**Flooding the entire Network with Initial Network Address:**

This is the part which is adopted from Link state Routing which enhances the routing in VIMNet to a greater efficient level. Flooding [14] makes the routers at each end of Virtual Circuits and in the entire network aware of the network and addresses of the routers in network. Though flooding consumes the resources these is very important and even the resource utilization, as it is one of the architectural advantages of VIMNet as it is deployed with high-speed optical fibers, which support duplex communications with full extent of the allocation bandwidth. The following figure depicts the flooding technique.

![Flooding Network Diagram](image)

Figure 4-9: Flooding of the network to routers in the network with addresses.
Packet is initiated at sender with destination router address

Packet is deployed at the sender with header similar to that of ATM as shown in figure 4.8

Now it contains the VCI which it should connect to and is equipped with CLP cell loss priority which decides which type of communication it is setting up a Real time packet or a Non real time packet. PT depicts the payload type. H count gives information for the router at the end of a virtual circuit, the network metric hop count so that it updates its routing table.

Each router at the end of Virtual Circuits updating the network graph

This step includes updating the network graph of the network with the distances from each router in the network by the updates it attains through the packet. Updating a router tables enables in making an efficient decision during its dispatch of packets to the network

Error handling and routing

Here in VIMNet as the probability of packet loss or errors in routing is minimal as the communication is connection oriented not connection less. Even though the packet is equipped with HEC a type of error checksum code that is checked at the end of route, which makes the network much efficient that, the present day networks.

4.4 Advantages of routing in VIMNet

VIMNet has its own advantages to its counter parts as the entire architecture is so closely connected with optical fibers and also by replacing the core set of routers with high powerful ATM switches. The routing has high beneficiary aspects such as effective
bandwidth utilization; VIMNet has lower packet loss and less probability of congestion, VIMNet if supplied with the presumed locale will be the answer for the issues and challenges that are associated with the present Internet architecture.
Chapter 5

Results and Conclusions

5.1 Results and inference from simulation

The VIMNet is simulated on a non-hardware platform as the hardware for the router is not yet specified even with theoretical assumptions for the router and with the underlying architecture of VIMNet the simulation is done in two different architectural scenarios.

The first architectural scenario is a connection less network which resembles the present Internet (on a broader scope) and the other is one with four closely connected routers which perform a 4X4 duplex mode of connection the test bed contains 25 routers in each case and the simulation screen shots and the results are shown below which prove the VIMNet is a better answer in both architectural and deployable mode.
Figure 5-1: The layout of routers arranged in Present Internet architecture

The following graphs depict the comparison of different metrics at the routers in the network with two layouts one as per the present Internet and another with VIMNet.

Figure 5-2: Average end-end delay in the layout as per the present Internet
Figure 5-3: Average end-end delay in the layout which is per VIMNet

Figure 5-4: Average jitter in the layout as per the present Internet
Figure 5-5: Average jitter in the layout as per the VIMNet

Figure 5-6: Last packet received in the layout as per the present Internet
Figure 5-7: Last packet received in the layout as per VIMNet

Figure 5-8: Throughput in the layout as per the present Internet
Figure 5-9: Throughput in the layout as per VIMNet

Figure 5-10: Total bytes received in the layout as per the present Internet
Figure 5-11: Total bytes received in the layout as per VIMNet

Figure 5-12: Total packet received in the layout as per the present Internet
Figure 5-13: Total packets received in the layout as per VIMNet

Figure 5-14: Bellman ford number of update sent in the layout as per the present Internet
Figure 5-15: Bellman ford no of updates sent in the layout as per VIMNet

Even with some tweaks here and there overall if we see VIMNet architecture is of better standard than the present Internet architecture. This makes the shuts the doors on the challenges and issues that are being faced in present Internet.

5.2 Future Work

Till now the implementation of VIMNet over the network has proved many advantages and left us questions unanswered that leads to future work as in the case of Implementing the VIMNet there should be a way to minimize the cost associated with its idea to overhaul the present network system which might be addressed with the growth in both in technology and nano-technology and also there is another scope for future work associated with adaptability to current Internet comes and the architecture of VIMNet is in such a way that data can be wrapped up or encrypted upon the end users choice so that the packet size will be adjusted such a way that can be used by the present Internet.
Still Implementations of modernized routing protocols can be done with the VIMNet and rate or adjust the hardware associated with VIMNet and it is assured that the Quality of service won’t drop in any of those experiments.
References

1. Recommendation I.150, B-ISDN Asynchronous Transfer Mode Functional Characteristics


4. X.200 : Information technology - Open Systems Interconnection - Basic Reference Model: The basic model

5. ITU-T X-Series Recommendations.

6. Routers Hold key to MPLS Measurement RFC 4377


11. RFC 3468


20. VIMNet by L Millerand JJ gruseney  IEEE DOI: 10.1109/EIT.2007.4374521

21. Wikipedia for reference to public standards such as ARPANET,X.25,MPLS,ATM

22. Cisco learning press for networking protocols
Appendix A

Source Code for Routing in VIMNet and Flooding based on DVR

A.1 Source Code for Routing in VIMNet

```c
#include <stdio.h>

typedef struct {
    int u, v, w;
} Edge;

int n; /* the number of nodes */
int e; /* the number of edges */
Edge edges[1024]; /* large enough for n <= 2^5 = 32 */
int d[32]; /* d[i] is the minimum distance from node s to node i */

#define INFINITY 10000

void printDist() {
    int i;

    printf("Distances:\n");

    for (i = 0; i < n; ++i)
        printf("to %d\t", i + 1);
    printf("\n");

    for (i = 0; i < n; ++i)
        printf("%d\t", d[i]);
    printf("\n\n");
}
```

57
void dvr(int s) {
    int i, j;
    for (i = 0; i < n; ++i)
        d[i] = INFINITY;
    d[s] = 0;
    for (i = 0; i < n - 1; ++i)
        for (j = 0; j < e; ++j)
            if (d[edges[j].u] + edges[j].w < d[edges[j].v])
}

int main(int argc, char *argv[]) {
    int i, j, w;
    FILE *fin = fopen("dist.txt", "r");
    fscanf(fin, "%d", &n);
    e = 0;
    for (i = 0; i < n; ++i)
        for (j = 0; j < n; ++j) {
            fscanf(fin, "%d", &w);
            if (w != 0) {
                edges[e].u = i;
                edges[e].v = j;
                edges[e].w = w;
                ++e;
            }
        }
    fclose(fin);
    /* printDist(); */
    dvr(0);
    printDist();
    return 0;
}
A.2 Source Code for configuring Network in VIMNet

#*******************Parallel Settings****************************************************

PARTITION-SCHEME AUTO

#*******************ATM Configuration****************************************************

DUMMY-ATM-LOGICAL-SUBNET-CONFIGURED YES
ATM-STATIC-ROUTE NO

#******************Dynamic Parameters****************************************************

DYNAMIC-ENABLED NO

#*****************************Terrain****************************************************

**

COORDINATE-SYSTEM CARTESIAN
TERRAIN-DIMENSIONS (1500, 1500)
DUMMY-ALTITUDES (1500, 0)
DUMMY-ENABLE-URBAN-TERRAIN-FEATURE NO
WEATHER-MOBILITY-INTERVAL 100MS

#******************Channel Properties****************************************************

PROPAGATION-CHANNEL-FREQUENCY[0] 2400000000
PROPAGATION-MODEL[0] STATISTICAL
PROPAGATION-PATHLOSS-MODEL[0] TWO-RAY
PROPAGATION-SHADOWING-MODEL[0] CONSTANT
PROPAGATION-SHADOWING-MEAN[0] 4.0
PROPAGATION-FADING-MODEL[0] NONE
PROPAGATION-LIMIT[0] -111.0
PROPAGATION-MAX-DISTANCE[0] 0
PROPAGATION-COMMUNICATION-PROXIMITY[0] 400
PROPAGATION-PROFILE-UPDATE-RATIO[0] 0.0

#******************Mobility and Placement****************************************************

NODE-PLACEMENT FILE
NODE-POSITION-FILE mohan.nodes
MOBILITY NONE

#**************************************************STATISTICS**************************************************
*****

PHY-LAYER-STATISTICS YES
MAC-LAYER-STATISTICS YES
ACCESS-LIST-STATISTICS NO
ARP-STATISTICS YES
ROUTING-STATISTICS YES
POLICY-ROUTING-STATISTICS NO
QOSPF-STATISTICS NO
ROUTE-REDISTRIBUTION-STATISTICS NO
EXTERIOR-GATEWAY-PROTOCOL-STATISTICS YES
NETWORK-LAYER-STATISTICS YES
INPUT-QUEUE-STATISTICS NO
INPUT-SCHEDULER-STATISTICS NO
QUEUE-STATISTICS YES
SCHEDULER-STATISTICS NO
SCHEDULER-GROUP-STATISTICS NO
DIFFSERV-EDGE-ROUTER-STATISTICS NO
ICMP-STATISTICS NO
IGMP-STATISTICS NO
NDP-STATISTICS NO
MOBILE-IP-STATISTICS NO
TCP-STATISTICS YES
UDP-STATISTICS YES
RSVP-STATISTICS NO
SRM-STATISTICS NO
RTP-STATISTICS NO
APPLICATION-STATISTICS YES
BATTERY-MODEL-STATISTICS NO
ENERGY-MODEL-STATISTICS YES
MOBILITY-STATISTICS NO
CELLULAR-STATISTICS YES
GSM-STATISTICS NO
VOIP-SIGNALLING-STATISTICS NO
SWITCH-PORT-STATISTICS NO
SWITCH-SCHEDULER-STATISTICS NO
SWITCH-QUEUE-STATISTICS NO
MPLS-STATISTICS NO
MPLS-LDP-STATISTICS NO
HOST-STATISTICS NO

#**************************************************PACKET TRACING**************************************************
PACKET-TRACE NO
ACCESS-LIST-TRACE NO

#******************Supplemental Files***********************************

APP-CONFIG-FILE satish.app
DUMMY-USER-PROFILE-FILE-NUMBER 0
DUMMY-TRAFFIC-PATTERN-FILE-NUMBER 0
DUMMY-ARBITRARY-DISTRIBUTION-FILE-NUMBER 0

#***********************HLA Interface***********************************

HLA NO

#***********************DIS Interface***********************************

DIS NO

#***********************STK Interface***********************************

STK-ENABLED NO

#**********************Physical Layer***********************************

PHY-LISTENABLE-CHANNEL-MASK 1
PHY-LISTENING-CHANNEL-MASK 1
PHY-MODEL PHY802.11b
PHY802.11-AUTO-RATE-FALLBACK NO
PHY802.11-DATA-RATE 2000000
PHY802.11b-TX-POWER--1MBPS 15.0
PHY802.11b-TX-POWER--2MBPS 15.0
PHY802.11b-TX-POWER--6MBPS 15.0
PHY802.11b-TX-POWER-11MBPS 15.0
PHY802.11b-RX-SENSITIVITY--1MBPS -94.0
PHY802.11b-RX-SENSITIVITY--2MBPS -91.0
PHY802.11b-RX-SENSITIVITY--6MBPS -87.0
PHY802.11b-RX-SENSITIVITY-11MBPS -83.0
PHY802.11-ESTIMATED-DIRECTIONAL-ANTENNA-GAIN 15.0
PHY-RX-MODEL PHY802.11b
ANTENNA-GAIN 0.0
ANTENNA-HEIGHT 1.5
ANTENNA-EFFICIENCY 0.8
ANTENNA-MISMATCH-LOSS 0.3
ANTENNA-CABLE-LOSS 0.0
ANTENNA-CONNECTION-LOSS 0.2
ANTENNA-MODEL OMNIDIRECTIONAL
PHY-TEMPERATURE 290.0
PHY-NOISE-FACTOR 10.0
ENERGY-MODEL-SPECIFICATION NONE

#***************************MAC Layer***************************

LINK-MAC-PROTOCOL ABSTRACT
LINK-PROPAGATION-DELAY 1MS
LINK-BANDWIDTH 10000000
LINK-HEADER-SIZE-IN-BITS 224
LINK-TX-FREQUENCY 13170000000
LINK-RX-FREQUENCY 13170000000
LINK-TX-ANTENNA-HEIGHT 30
LINK-RX-ANTENNA-HEIGHT 30
LINK-TX-ANTENNA-DISH-DIAMETER 0.8
LINK-RX-ANTENNA-DISH-DIAMETER 0.8
LINK-TX-ANTENNA-CABLE-LOSS 1.5
LINK-RX-ANTENNA-CABLE-LOSS 1.5
LINK-TX-POWER 30
LINK-RX-SENSITIVITY -80
LINK-NOISE-TEMPERATURE 290
LINK-NOISE-FACTOR 4
LINK-TERRAIN-TYPE PLAINS
LINK-PROPAGATION-RAIN-INTENSITY 0
LINK-PROPAGATION-TEMPERATURE 25
LINK-PROPAGATION-SAMPLING-DISTANCE 100
LINK-PROPAGATION-CLIMATE 1
LINK-PROPAGATION-REFRACTIVITY 360
LINK-PROPAGATION-PERMITTIVITY 15
LINK-PROPAGATION-CONDUCTIVITY 0.005
LINK-PROPAGATION-HUMIDITY 50
LINK-PERCENTAGE-TIME-REFRACTIVITY-GRADIENT-LESS-STANDARD 15

MAC-PROTOCOL MACDOT11
MAC-DOT11-SHORT-PACKET-TRANSMIT-LIMIT 7
MAC-DOT11-LONG-PACKET-TRANSMIT-LIMIT 4
MAC-DOT11-RTS-THRESHOLD 0
MAC-DOT11-STOP-RECEIVING-AFTER-HEADER-MODE NO
MAC-DOT11-ASSOCIATION NONE
MAC-DOT11-IBSS-SUPPORT-PS-MODE NO
MAC-DOT11-DIRECTIONAL-ANTENNA-MODE NO
MAC-SECURITY-PROTOCOL NO
WORMHOLE-VICTIM-COUNT-TURNAROUND-TIME NO
MAC-PROPAGATION-DELAY 1US

#***************************Schedulers and Queues***********************************
IP-QUEUE-PRIORITY-INPUT-QUEUE-SIZE 150000
IP-QUEUE-SCHEDULER STRICT-PRIORITY
IP-QUEUE-NUM-PRIORITIES 3
IP-QUEUE-PRIORITY-QUEUE-SIZE 150000
QUEUE-WEIGHT 0
IP-QUEUE-TYPE FIFO

#***************************QoS Configuration***********************************

#***************************Network Security***********************************

#***************************ROUTER MODEL***********************************

#***************************NETWORK LAYER***********************************
#ROUTING PROTOCOL

ROUTING-PROTOCOL BELLMANFORD
STATIC-ROUTE NO
DEFAULT-ROUTE NO
HSRP-PROTOCOL NO

#TRANSPORT

TRANSPORT-PROTOCOL-RSVP YES
GUI_DUMMY_CONFIG_TCP YES
TCP LITE
TCP-USE-RFC1323 NO
TCP-DELAY-SHORT-PACKETS-ACKS NO
TCP-USE-NAGLE-ALGORITHM YES
TCP-USE-KEEPALIVE-PROBES YES
TCP-USE-OPTIONS YES
TCP-DELAY-ACKS YES
TCP-MSS 512
TCP-SEND-BUFFER 16384
TCP-RECEIVE-BUFFER 16384

#MPLS Specs

MPLS-PROTOCOL NO

#Application Layer

DUMMY-VOIP-APPLICATION-EXISTS NO
RTP-ENABLED NO

#USER BEHAVIOR

DUMMY-UBEE-ENABLED NO

#Battery Models

BATTERY-MODEL NONE
# Adaptation Protocol

ADAPTATION-PROTOCOL AAL5
ATM-CONNECTION-REFRESH-TIME 5M
ATM-CONNECTION-TIMEOUT-TIME 1M

### [Default Wireless Subnet]

SUBNET N8-169.0.0.0 {1, 2, 4 thru 25} Default

### Physical Layer

[ N8-169.0.0.0 ] PHY-LISTENABLE-CHANNEL-MASK[0] 1
[ N8-169.0.0.0 ] PHY-LISTENING-CHANNEL-MASK[0] 1
[ N8-169.0.0.0 ] PHY-MODEL PHY802.11b
[ N8-169.0.0.0 ] PHY802.11-AUTO-RATE-FALLBACK NO
[ N8-169.0.0.0 ] PHY802.11-DATA-RATE 2000000
[ N8-169.0.0.0 ] PHY802.11b-TX-POWER--1MBPS 15.0
[ N8-169.0.0.0 ] PHY802.11b-TX-POWER--2MBPS 15.0
[ N8-169.0.0.0 ] PHY802.11b-TX-POWER--6MBPS 15.0
[ N8-169.0.0.0 ] PHY802.11b-TX-POWER-11MBPS 15.0
[ N8-169.0.0.0 ] PHY802.11b-RX-SENSITIVITY--1MBPS -94.0
[ N8-169.0.0.0 ] PHY802.11b-RX-SENSITIVITY--2MBPS -91.0
[ N8-169.0.0.0 ] PHY802.11b-RX-SENSITIVITY--6MBPS -87.0
[ N8-169.0.0.0 ] PHY802.11b-RX-SENSITIVITY-11MBPS -83.0
[ N8-169.0.0.0 ] PHY802.11-ESTIMATED-DIRECTIONAL-ANTENNA-GAIN 15.0
[ N8-169.0.0.0 ] PHY-RX-MODEL PHY802.11b
[ N8-169.0.0.0 ] ANTENNA-GAIN 0.0
[ N8-169.0.0.0 ] ANTENNA-HEIGHT 1.5
[ N8-169.0.0.0 ] ANTENNA-EFFICIENCY 0.8
[ N8-169.0.0.0 ] ANTENNA-MISMATCH-LOSS 0.3
[ N8-169.0.0.0 ] ANTENNA-CABLE-LOSS 0.0
[ N8-169.0.0.0 ] ANTENNA-CONNECTION-LOSS 0.2
[ N8-169.0.0.0 ] ANTENNA-MODEL OMNIDIRECTIONAL

### Network Layer

[ N8-169.0.0.0 ] NETWORK-PROTOCOL IP

### Interface Configuration
[1] NETWORK-PROTOCOL[0]       IP
[1] IP-SUBNET-MASK[0]         255.255.255.0
[1] IP-ADDRESS[0]             169.0.0.1

[2] NETWORK-PROTOCOL[0]       IP
[2] IP-SUBNET-MASK[0]         255.255.255.0
[2] IP-ADDRESS[0]             169.0.0.2

[4] IP-SUBNET-MASK[0]         255.255.255.0
[4] IP-ADDRESS[0]             169.0.0.4

[5] NETWORK-PROTOCOL[0]       IP
[5] IP-SUBNET-MASK[0]         255.255.255.0
[5] IP-ADDRESS[0]             169.0.0.5

[6] NETWORK-PROTOCOL[0]       IP
[6] IP-SUBNET-MASK[0]         255.255.255.0
[6] IP-ADDRESS[0]             169.0.0.6

[7] NETWORK-PROTOCOL[0]       IP
[7] IP-SUBNET-MASK[0]         255.255.255.0
[7] IP-ADDRESS[0]             169.0.0.7

[8] NETWORK-PROTOCOL[0]       IP
[8] IP-SUBNET-MASK[0]         255.255.255.0
[8] IP-ADDRESS[0]             169.0.0.8

[9] NETWORK-PROTOCOL[0]       IP
[9] IP-SUBNET-MASK[0]         255.255.255.0
[9] IP-ADDRESS[0]             169.0.0.9

[10] NETWORK-PROTOCOL[0]      IP
[10] IP-SUBNET-MASK[0]        255.255.255.0
[10] IP-ADDRESS[0]            169.0.0.10

<table>
<thead>
<tr>
<th>Network-Protocol</th>
<th>IP-Subnet-Mask</th>
<th>IP-Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP</td>
<td>255.255.255.0</td>
<td>169.0.0.11</td>
</tr>
<tr>
<td>IP</td>
<td>255.255.255.0</td>
<td>169.0.0.12</td>
</tr>
<tr>
<td>IP</td>
<td>255.255.255.0</td>
<td>169.0.0.13</td>
</tr>
<tr>
<td>IP</td>
<td>255.255.255.0</td>
<td>169.0.0.14</td>
</tr>
<tr>
<td>IP</td>
<td>255.255.255.0</td>
<td>169.0.0.15</td>
</tr>
<tr>
<td>IP</td>
<td>255.255.255.0</td>
<td>169.0.0.16</td>
</tr>
<tr>
<td>IP</td>
<td>255.255.255.0</td>
<td>169.0.0.17</td>
</tr>
<tr>
<td>IP</td>
<td>255.255.255.0</td>
<td>169.0.0.18</td>
</tr>
<tr>
<td>IP</td>
<td>255.255.255.0</td>
<td>169.0.0.19</td>
</tr>
<tr>
<td>IP-SUBNET-MASK [0]</td>
<td>255.255.255.0</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td>IP-ADDRESS [0]</td>
<td>169.0.0.20</td>
<td></td>
</tr>
<tr>
<td>NETWORK-PROTOCOL [0]</td>
<td>IP</td>
<td></td>
</tr>
<tr>
<td>IP-SUBNET-MASK [0]</td>
<td>255.255.255.0</td>
<td></td>
</tr>
<tr>
<td>IP-ADDRESS [0]</td>
<td>169.0.0.21</td>
<td></td>
</tr>
<tr>
<td>NETWORK-PROTOCOL [0]</td>
<td>IP</td>
<td></td>
</tr>
<tr>
<td>IP-SUBNET-MASK [0]</td>
<td>255.255.255.0</td>
<td></td>
</tr>
<tr>
<td>IP-ADDRESS [0]</td>
<td>169.0.0.22</td>
<td></td>
</tr>
<tr>
<td>NETWORK-PROTOCOL [0]</td>
<td>IP</td>
<td></td>
</tr>
<tr>
<td>IP-SUBNET-MASK [0]</td>
<td>255.255.255.0</td>
<td></td>
</tr>
<tr>
<td>IP-ADDRESS [0]</td>
<td>169.0.0.23</td>
<td></td>
</tr>
<tr>
<td>NETWORK-PROTOCOL [0]</td>
<td>IP</td>
<td></td>
</tr>
<tr>
<td>IP-SUBNET-MASK [0]</td>
<td>255.255.255.0</td>
<td></td>
</tr>
<tr>
<td>IP-ADDRESS [0]</td>
<td>169.0.0.24</td>
<td></td>
</tr>
<tr>
<td>NETWORK-PROTOCOL [0]</td>
<td>IP</td>
<td></td>
</tr>
<tr>
<td>IP-SUBNET-MASK [0]</td>
<td>255.255.255.0</td>
<td></td>
</tr>
<tr>
<td>IP-ADDRESS [0]</td>
<td>169.0.0.49</td>
<td></td>
</tr>
<tr>
<td>PHY-LISTENABLE-CHANNEL-MASK</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>PHY-LISTENING-CHANNEL-MASK</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>IP-QUEUE-TYPE [0]</td>
<td>FIFO</td>
<td></td>
</tr>
<tr>
<td>IP-QUEUE-PRIORITY-QUEUE-SIZE [0]</td>
<td>150000</td>
<td></td>
</tr>
<tr>
<td>IP-QUEUE-PRIORITY-QUEUE-SIZE [0]</td>
<td>150000</td>
<td></td>
</tr>
</tbody>
</table>
69.0.0.18 69.0.0.19 69.0.0.20 69.0.0.21 69.0.0.22 69.0.0.23 69.0.0.24 69.0.0.49] IP-QUEUE-PRIORITY-QUEUE-SIZE[2] 150000
[69.0.0.1 69.0.0.2 69.0.0.4 69.0.0.5 69.0.0.6 69.0.0.7 69.0.0.8 69.0.0.9 69.0.0.10 69.0.0.11 69.0.0.12 69.0.0.13 69.0.0.14 69.0.0.15 69.0.0.16 69.0.0.17 69.0.0.18 69.0.0.19 69.0.0.20 69.0.0.21 69.0.0.22 69.0.0.23 69.0.0.24 69.0.0.49] IP-QUEUE-PRIORITY-QUEUE-SIZE[1] 150000
[69.0.0.1 69.0.0.2 69.0.0.4 69.0.0.5 69.0.0.6 69.0.0.7 69.0.0.8 69.0.0.9 69.0.0.10 69.0.0.11 69.0.0.12 69.0.0.13 69.0.0.14 69.0.0.15 69.0.0.16 69.0.0.17 69.0.0.18 69.0.0.19 69.0.0.20 69.0.0.21 69.0.0.22 69.0.0.23 69.0.0.24 69.0.0.49] IP-QUEUE-PRIORITY-QUEUE-SIZE[1] 150000
[69.0.0.1 69.0.0.2 69.0.0.4 69.0.0.5 69.0.0.6 69.0.0.7 69.0.0.8 69.0.0.9 69.0.0.10 69.0.0.11 69.0.0.12 69.0.0.13 69.0.0.14 69.0.0.15 69.0.0.16 69.0.0.17 69.0.0.18 69.0.0.19 69.0.0.20 69.0.0.21 69.0.0.22 69.0.0.23 69.0.0.24 69.0.0.49] IP-QUEUE-TYPE[2] FIFO

#***************Hierarchy Configuration****************************************************

#******************Node Configuration********************************************************

[5 thru 24] MOBILITY-POSITION-GRANULARITY 1.0
[1 2] NETWORK-PROTOCOL IP
[4 thru 25] NETWORK-PROTOCOL IP
[1 2] IP-LOOPBACK-ADDRESS 127.0.0.1
[4 thru 25] IP-LOOPBACK-ADDRESS 127.0.0.1
[5 thru 24] MOBILITY-WP-MIN-SPEED 0
[5 thru 24] MOBILITY RANDOM-WAYPOINT
[1] HOSTNAME host1
[2] HOSTNAME host2
[4] HOSTNAME host4
[5] HOSTNAME host5
[6] HOSTNAME host6
[7] HOSTNAME host7
[8] HOSTNAME host8
[9] HOSTNAME host9
[10] HOSTNAME host10
[12] HOSTNAME host12
[13] HOSTNAME host13
[14] HOSTNAME host14
[15] HOSTNAME host15
[16] HOSTNAME host16
[17] HOSTNAME host17
A.3 Source Code for throughput in the network architecture

#!/user/bin/perl

use warnings;

use strict;

my $stat_file = $ARGV[0];

unless ($stat_file){
  die "unable to load the file : $!";
}

open (STAT,"$stat_file") or die "can not open the file : $!";

open (STAT_OUT,">>out_stat.stat") or die "can not create output file : $!";

# print "Enter the node id to find its throughput : ";

# my $node_id = <STDIN>
#chomp($node_id);
#print "node : $node_id\n";
my $flag = 0;
my $Th_sum = 0;
my $percent_sum = 0;

foreach my $line (<STAT>){
    my @units = split(/,/,$line);
    my $sub_line = $units[5];
    my $sub_line2 = $units[4];
    chomp($sub_line);
    chomp($sub_line2);
    my @sub_units = split(/ /,$sub_line);
    my $len = @sub_units;
    #print "length : $len\n";
    if($line =~ m/Server/ && $line =~ m/Throughput/ && $units[0] <= 100){
        my $thruput = $sub_units[$len-1];
        $Th_sum = $Th_sum + $thruput;
        print "Node $units[0] throughput = $thruput bps\n";
        $flag = 1;
    }
}

#if(!$flag){
    # print "there is no record for Node $node_id\n";
}
# print STAT_OUT "No record for node $node_id\n";
#
}

$percent_sum = $Th_sum / 1000000;

print "Total throughput = $Th_sum bps\n";
print "Percentage throughput = $percent_sum Mbps\n";
print STAT_OUT "$percent_sum\n"
close(STAT);
close (STAT_OUT);