AN ENHANCED INFORMATION CENTRIC MODEL FOR EFFICIENT BANDWIDTH MANAGEMENT

JAMES GITONGORI OMBOGO

MASTER OF SCIENCE

(Computer Systems)

JOMO KENYATTA UNIVERSITY OF AGRICULTURE AND TECHNOLOGY

2015

An Enhanced Information Centric Model for Efficient Bandwidth Management

James Gitongori Ombogo

A Research Thesis submitted to the School of Computing and Information Technology in Partial Fulfillment of the requirements for the award of the degree of Master of Science in Computer Systems in the Jomo Kenyatta University of Agriculture and Technology

2015

DECLARATION

This thesis is my original work and has not been presented for a degree in any other university.

Signature: _____ Date: _____

JAMES GITONGORI OMBOGO

This thesis has been submitted for examination with our approval as the university supervisors:

SIGNATURE_____ DATE_____

Dr. Cheruiyot W.K,

JKUAT, Kenya

SIGNATURE_____ DATE_____

Mr. Sylvester Kiptoo,

JKUAT, Kenya

DEDICATION

My dedication goes to my God for giving me life, being my sustainer and for blessing me academically to realize long term dream. Not forgetting my parents Joseph Ombogo , Rael Moraa Ombogo, my wife Caren Mayabi and my daughter Gracia Moraa Gitongori for their inspiration, support and encouragement that always made me to keep hope alive. May they find wisdom and knowledge in my realms.

ACKNOWLEDGEMENT

I am very grateful for the support and guidance of the following people and institutions during my studies without which this research would not have been completed successfully. I am grateful to my supervisors Dr. Wilson Cheruiyot and Mr. Sylvester Kiptoo for their constant guidance, critical appraisals, motivation, valuable suggestions and supervision, which has made the completion of this thesis. Despite their busy schedule they made time to meet me and to go through my work.

I am also very thankful to Jomo Kenyatta University of Agriculture and Technology for granting me a chance and a conducive environment to study at their institution. My special thanks also goes to the following persons for their support and motivation towards my research thesis in various ways; Dr. Agnes Mindila, Cyrus Abanti, Mr. Joseph Ombogo and the masters class 2013 of Computer Systems Kisii CBD campus.

Finally, I wish to acknowledge my wife, Caren Mayabi for the love she brings to my life and for her intellectual support. Her zest and pursuit of excellence are a driving force; lots of Love!.Above all I thank the Almighty God for His everlasting love and protection.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACK	Acknowledgment
API	Application Programming Interface
CCN	Content Centric Network
COMET	Content Mediation Network
CR	Content Router
CS	Content Store
DHT	Distributed Hash Table
DONA	Data Oriented Network Architecture
ENDRE	End Redundancy Elimination
ICM	Information Centric Model
ICN	Information Centric Networking
IP	Internet Protocol
ISP	Internet Service Providers
NDO	Named Data Object
NDN	Named Data Networking
NetInf	NETwork of Information
NRS	Named Resolution System
PPPOE	Point to Point Protocol Over the Ethernet
PURSUIT	PUblish Subscribe archITechture
RE	Redundancy Elimination
SYNC	Synchronization
TCP	Transport Control Protocol

ABSTRACT

Information centric model is a new approach to networking which enables efficient network software-based control and application independent information caching. However, in a transmission network, there exist redundant bytes in packets that are cached at content routers hence exhausting bandwidth and occasioning such problems as bandwidth glitches, low throughput, and denial of service among others. This study developed an enhanced redundancy elimination mechanism which takes into account minimal network memory operations leading to optimum bandwidth management and network power consumption reduction. The main aim of the research was to come up with an enhanced information centric model to identify and eliminate redundancy so as to mitigate network glitches such as bandwidth glitches, low throughput, and denial of service among others. The research used purpose sampling to select sample network sites, collected data and evaluated our model using both real time network traffic data and synthetic traffic. The research analyzed bytes in packets and found out that there were 46% redundant bytes. The research compared various file types in transit and found that 97% of traffic was binary files such as video, audio and pictures while 3% constituted text files. This prompted the research to find out file organization of binary files which was found that they hardly partially intersect therefore was no need for chunking them. It was found that available bandwidth will be optimized by eliminating redundant bytes and reducing number of shim headers. Implementation of the enhanced bandwidth management model will make internet applications be more reliable due to optimized bandwidth and reduced processing load.

CHAPTER ONE

1.0 Introduction

In the world today, computer networks are playing a central role in work, business, education, entertainment and social life. Organizations and individuals are benefiting greatly as computer networks have made communication faster and reliable through applications such as emails and video conferencing among others. Moreover, Tanenbaum and Wetherall (2011) argues that nowadays it is easier to control machines remotely, share peripheral devices, data and programs. Therefore it is important to efficiently manage the way data is transmitted over the network in order not to overload the available bandwidth. Halgren (2012) agrees in his study that even though the capacity and speed of the network are constantly increasing and its associated costs are declining, it is still not a good reason for users to ignore the additional investments and efforts needed to optimize bandwidth management.

More studies reveal that investments in bandwidth optimization are the ones that can contribute to a reduction in total cost of ownership, specifically in respect to efficiency gains and maximized resources (Martin, 2010; Munir, 2014). This research has discussed more about bandwidth management which this chapter addresses and derives objectives that are used to achieve the solution to the problem.

1.1 Background of the Study

The Internet interconnects millions of computers, providing a global communication, storage, and computation infrastructure. Moreover, the Internet is currently being integrated with mobile and wireless technology, ushering in an impressive array of new applications (Kurose & Ross, 2013). For instance, Bonaventure (2011) has pointed out that internet has allowed distributed applications such as remote login, electronic mail, Web surfing, instant messaging, audio, video streaming, Internet telephony, distributed games, peer-to-peer (P2P) file sharing, and much more running on its end systems to exchange data with each other.

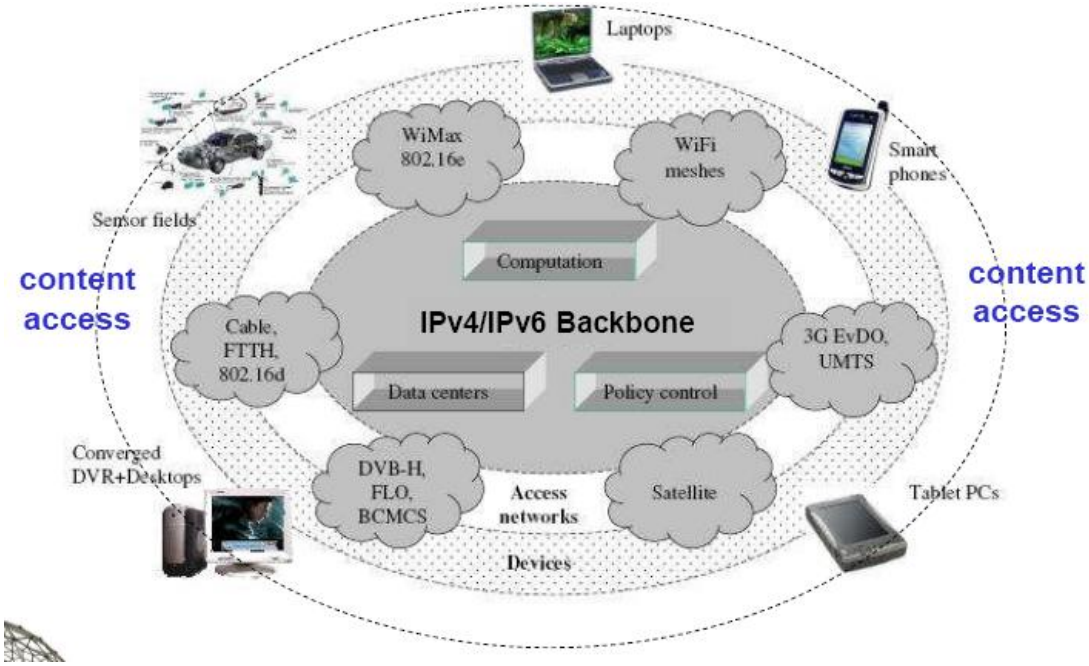


Figure 1.0: Internet Architecture. (Source: Pavlou, 2011, p.3)

This end system must communicate either through wired or wireless medium which has a laid down internet protocol, a good example is shown in figure 1.0. For instance, protocols in routers determine a packet's path from source to destination; hardware-implemented protocols in the network interface cards of two physically connected computers control the flow of bits on the wire between the two network interface cards; congestion-control protocols in end systems control the rate at which packets are transmitted between sender and receiver (Pavlou, 2011).

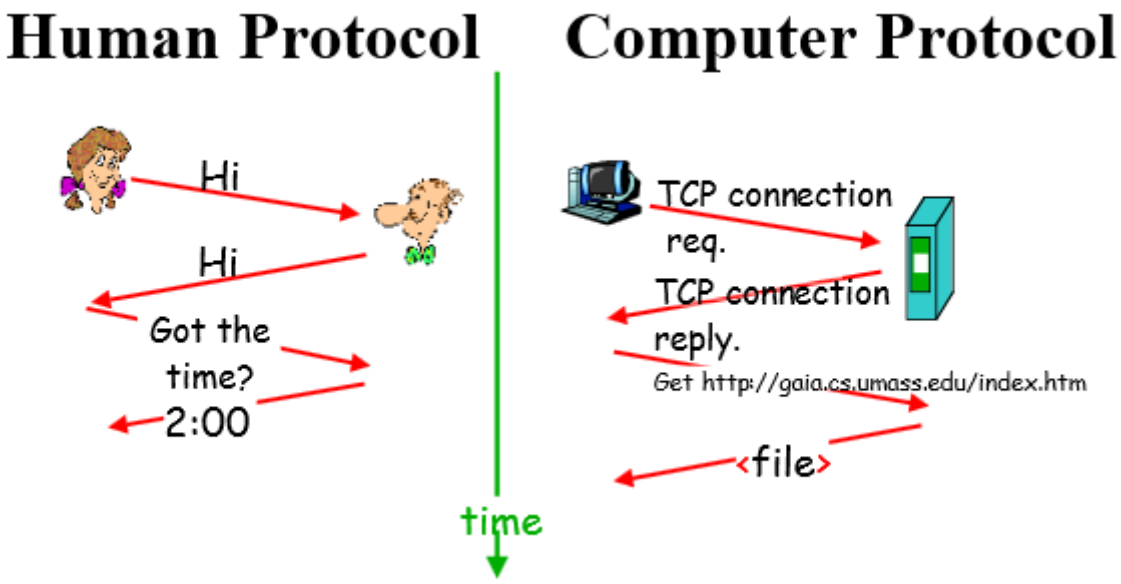


Figure 1.1 Human and Network Protocol. (Source: Tanenbaum&Wetherall, 2011, p.6)

Communication between sender and receiver can be easily understood using human analogies, since us humans communicate all of the time. Consider what you do when you want to ask someone for the time of day. A typical communication between two human being is shown in Figure 1.1. This analogy has been borrowed in computer protocols and has been used to set rules for network communication. In this transmission, long messages of data are broken into smaller messages called packets and routed to their destination (Tanenbaum & Wetherall, 2011).

Kurose and Ross (2010) have discussed more on network routing mechanisms in which they point out that each of packets in a transmission media needs to traverse from source to destination over routers and communication links. Packet routing approach has been used to ensure successful TCP client-server communication by involving a three way handshake between the sender and receiver before TCP connection is established (Schmidt, 2008) as shown in figure 2.1. Once a packet is received in its destination, its kept temporary in content router's cache before it's erased from memory (Kurose & Ross, 2010).

In particular, routing techniques have been evolving in order to adjust to the current networking bandwidth optimization needs (Bonaventure, 2011). Like in early packet routing approaches, Media Access addresses (MAC) of routers was used to uniquely identify packet source and destination (Kurose & Ross 2010; Stalling, 2007). However current routing approaches have shifted to information centric networking where information is viewed as first class entity of a network infrastructure (Parvlou, 2011). This has made it possible for network applications such as Bit torrent, YouTube, Google Video, Over the top Video, social networks and photo sharing sites to be more scalable, resilient and efficient (Tarnauca, 2011).

Despite major shift from host to host to information centric networks there is still network congestions due to overfilling network links to capacity (Halgren, 2012). This is due to rise of the amount of data travelling on the network. For this reason, investments in bandwidth optimization techniques have concentrated on techniques that can contribute to a reduction in total cost of ownership, especially in respect to efficiency gains and maximized network resources (Munir, 2004).

In many business organization all over the world, bandwidth is treated as a critical resource which need managed. In the recent past many researchers have experimented on Bandwidth management techniques such as throttling, middle boxes Redundancy Elimination have been suggested (Paulo, 2010; Munir, 2014) in a bid to manage bandwidth resource. For instance, Munir(2014) explains bandwidth throttling as an intentional limiting of network resource to users in an attempt to regulate network traffic and minimize network congestion. On the other hand Paulo (2010) point out that Middle box devices are used to analyze and reduce redundancy of incoming and outgoing content in networks. But these techniques can cause denial of service or sources of network failure (Paulo, 2010) hence the need for an advanced mechanism to mitigate the problems experienced by the current methods (Stoylar & Alexander, 2010). For example, in Kenyan universities, research and collaborative work are becoming regular processes prompting network administrators to manage bandwidth as a critical resource that is limited, expensive and of high demand (Martin, 2010).

1.2 Statement of the Problem

In a telecommunication network, there is duplication of bytes that are common to different packets cached, routed and forwarded over information centric network leading to bandwidth wastage and reduction of quality of service. Current network setups use bandwidth throttling, middleboxes, packet aware routing Redundant Elimination and byte level redundancy elimination mechanisms for bandwidth resource management. Bandwidth throttling blocks some links in networks in order to conserve bandwidth leading to denial of service, slow loading, skipping and stuttering experiences. On the other hand, Middlebox redundancy elimination devices are expensive, complex to manage, and creates new failure modes for the networks that use them. Constructing redundancy aware routes is challenging since it is not be economically viable to deploy redundancy elimination in every link. Also to preserve end to end performance and control signaling costs, routes cannot be determined on a per link basis, routes have to be determined independent of packet content. Byte level redundancy Elimination mechanism currently used in most of information centric networks is saves more bandwidth than packet aware routing RE but it has high processing demand due to high rate of processor cycles involved during chunking process.

Therefore this research has developed an enhanced information centric model that detects and eliminates redundant bytes in packets cached in routers, while reducing cost of memory operations. This ensures that there is no overfilling of the network links to capacity reducing such problems as denial of service, deadlocks and bottlenecks.

1.3 Problem Justification

It is desirable to introduce redundancy elimination mechanism in order to reduce network costs incurred through initial setup costs experienced when buying middle box devices. On the other hand, redundant bytes eliminated will reduce wastage of bundles purchased from ISP. In addition network throughput will be high as compared to the current network setups as more packets will be delivered at a given time frame making applications such as e-banking, e-commerce and video conferencing to be relied upon by end users. Furthermore, network and system administrators will benefit a lot by implementing this mechanism as one of the bandwidth management mechanism.

1.4 Objectives of the Study

The general objective of the study was to research on existing bandwidth management techniques in order to develop an enhanced information centric model for efficient bandwidth management in telecommunication networks.

The specific objectives of the study were to:

i. Analyze the network routing and caching optimization needs.

ii. Investigate current network content redundancy elimination techniques used in bandwidth optimization.

iii. Develop an enhanced information centric model that ensures efficient bandwidth management and reduces network processing load based on traffic type.

iv. Evaluate the performance of the enhanced information centric model by measuring percentage of bandwidth saving and processing load that it will provide as compared to existing model.

1.5 Research Questions

The following research questions guided the study:

i. What are the network routing and caching optimization needs?

ii. What are the network content redundancy elimination techniques currently used in bandwidth management?

iii. How is our redundancy aware routing model better than existing approaches in terms of bandwidth utilization, processing load and throughput?

What percentage of bandwidth saving our enhanced model provides compared to existing model.

1.6 Research Thesis organization

In order to meet the objectives defined in section 1.4, a systematic approach to analysis, design and comparison of the algorithms has been adopted. This thesis is organized into five chapters and a summary of all the chapters is given in Figure 1.2.

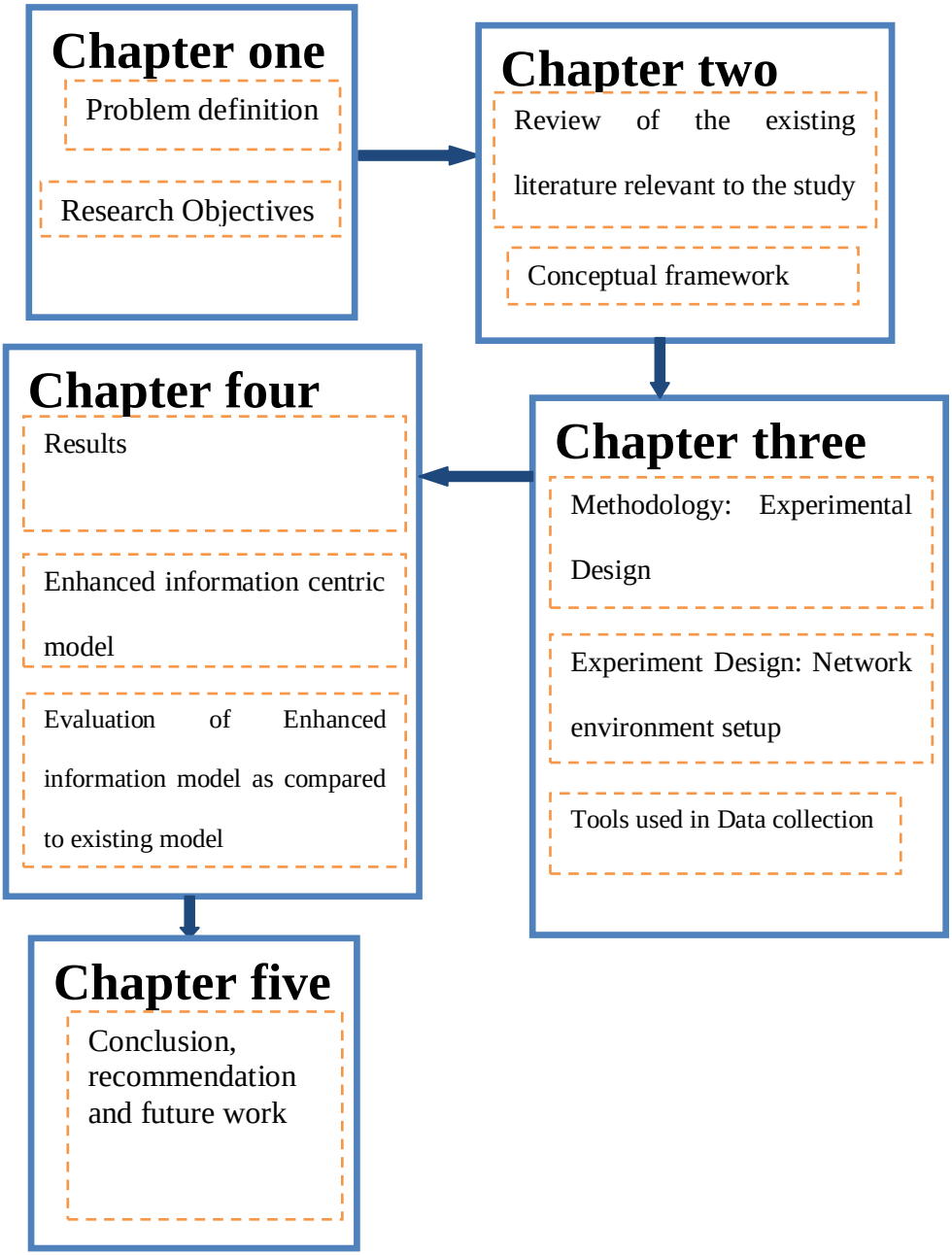


Figure 1.2: Research Thesis flow chart

CHAPTER TWO

LITERATURE REVIEW

2.0 Introduction

This study relied on knowledge from data communication especially as it applies to network protocols, switching, caching, bandwidth optimization and Information Centric designs. To the working of network protocols, the architecture of network protocol, packet switching and caching mechanisms, bandwidth optimization techniques and efficiency of architecture of information centric designs must be understood.

Computer networks exist to provide connectivity among a variety of computers and access devices. This connectivity provides means for encoded data to be transmitted from one location to another a general term referred as data communication (Kurose & Ross, 2013). Stalling (2007) has provided a data communication model to generalize the procedure involved in data communication. In his transmission model, the Source generates data to be transmitted, transmitter converts data into transmittable signals, transmission system carries data, Receiver converts received signal into data and destination takes incoming data as shown in figure 2.0.

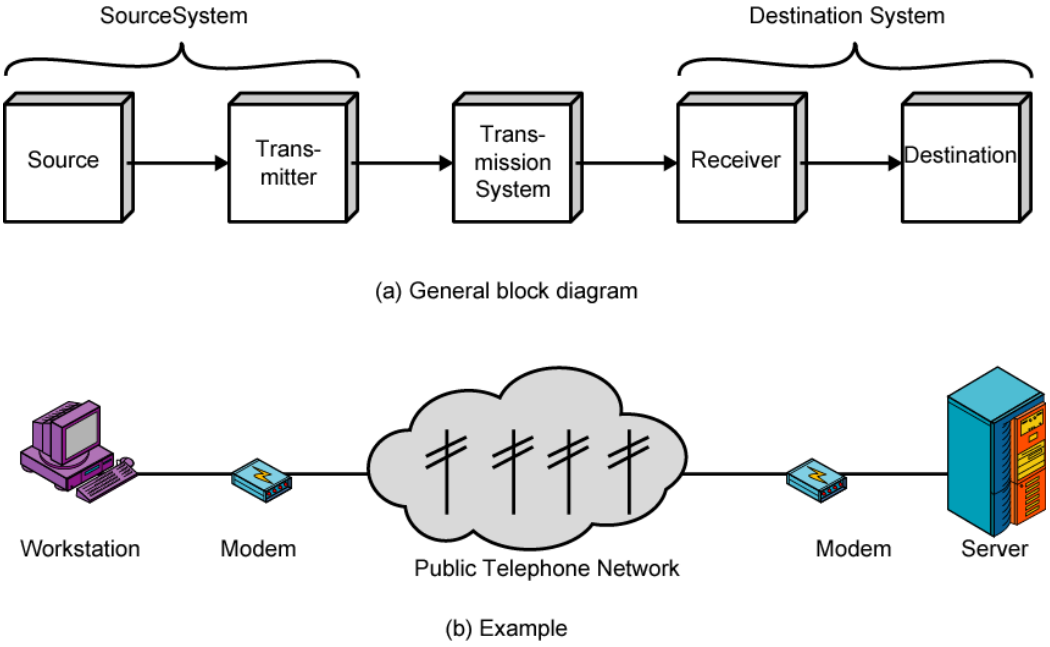


Figure 2.0: Data communication Model. (Source: Stalling, 2007, p.16)

Based on the data transmission model, orderly communication over a network must be ensured and all the nodes in the network must follow a set of rules called protocols. These rules are complex as they extend from the electric connection to the network and the format of the message, all the way to the interaction between application programs that run on different nodes (Tanenbaum & Wetherall, 2003). Bonaventure (2011) agrees in his study that for this communication to be effective, long messages of data are broken into smaller messages called packets for easier routing over the network channel and every channel has its own packet carrying capacity which if exceeded will lead to deadlocks and network failure. Therefore there's a need to measure channel carrying capacity which Munir (2014) expresses in terms of bandwidth.

2.1 Computer Network service and Protocol Architecture

Computer networks are becoming part and parcel of the world of today in terms of business, communication and entertainment (Bichanga & Wario, 2014). For computer devices connected in the network to work correctly, protocols have to be identified defining on how these electronic devices should be connected, and how data should be transmitted between them (Bonaventure, 2011). According to Kurose and Ross (2013), the internet is utilizing both wired (guided) and wireless (unguided) telecommunication link to provide electronic transmission of information over the internet. The information may be in the form of voice telephone calls, data, text, images, or video (Bonaventure, 2011).

The internet provides two types of services to its application: connection-oriented and connectionless services. When application uses connection oriented services, control packets must be send to each other before sending real data a procedure called handshaking. This handshaking procedure alerts the client and server, allowing them to prepare for an onslaught of packets. (Choi, 2010).

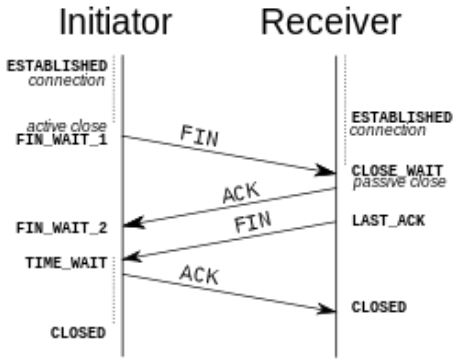


Figure 2.1: TCP three-way handshake connection process. (Source: Tanenbaum& Wetherall, 2011, p.6)

Transport Control Protocol is a good example of connection oriented protocol. To establish a connection, TCP uses a three-way handshake (Tanenbaum & Wetherall, 2011). Before a client attempts to connect with a server, the server must first bind to and listen at a port to open it up for connections: this is called a passive open (Stalling, 2007). Once the passive open is established, a client may initiate an active open. To establish a connection, the three-way handshake occurs: SYN: The active open is performed by the client sending a SYN to the server. The client sets the segment's sequence number to a random value A; SYN-ACK: In response, the server replies with a SYN-ACK (Schmid, 2008). The acknowledgment number is set to one more than the received sequence number i.e. A+1, and the sequence number that the server chooses for the packet is another random number, B; ACK: Finally, the client sends an ACK back to the server. The sequence number is set to the received acknowledgement value i.e. A+1, and the acknowledgement number is set to one more than the received sequence number i.e. B+1(Koponen & Chawla 2007).

Studies conducted by kurose and Ross (2013), Tanenbaum and Wetherall (2011) and Stalling (2007) reveal that the Internet's connection oriented service ensures that there is reliable data transfer, flow of control and congestion control. Reliable data transfer means that an application can rely on the connection to deliver all of its data without

In connectionless service, there is no handshaking meaning when one side of an application wants to send packets to another side of an application, the sending application simply sends the packets (Kurose & Ross, 2013). Since there is no handshaking procedure prior to the transmission of the packets, data can be delivered faster. But there are no acknowledgments either, so a source never knows for sure which packets arrive at the destination. Moreover, the service makes no provision for flow control or congestion control (Halgren 2012).

The Internet's connectionless service is provided by UDP (User Datagram Protocol); UDP is defined in the Internet Request for Comments RFC 768 [RFC 768]. Most of the more familiar Internet applications use TCP, the Internet's connection-oriented service. These applications include Telnet (remote login), SMTP (for electronic mail), FTP (for file transfer), and HTTP (for the Web). Nevertheless, Bonaventure (2011) concurs with Stalling (2007) that UDP, the Internet's connectionless service, is used by many applications, including many of the emerging multimedia applications, such as Internet phone, audio-on demand, and video conferencing (Martin 2010).

Kurose and Ross (2013) have matched connection and connectionless services into the reference models due to the growing complexity of computer networks to facilitate the description of network protocols and services. Of these, the Open Systems Interconnection model is probably the most influential (Stalling 2007). It serves as the basis for the standardization work performed within the ISO to develop global computer network standards and TCP/IP protocol architecture as shown in figure 2.2.

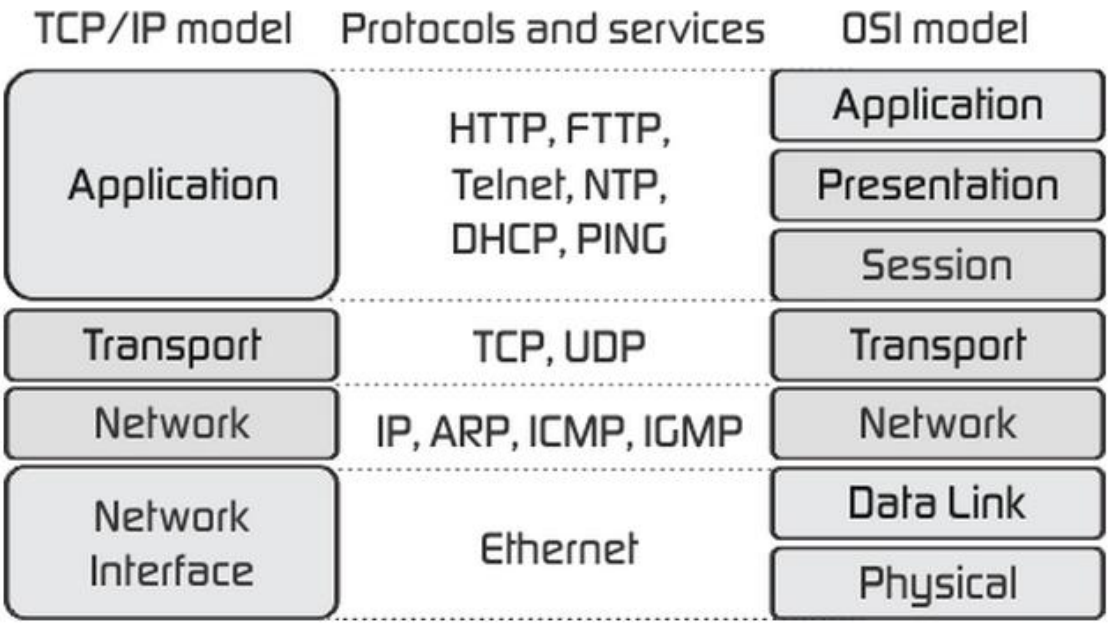


Figure 2.2: TCP/IP and OSI Model. (Source: Stalling, 2007, p.9).

The Open Systems Interconnection (OSI) reference model was developed by the International Organization for Standardization (ISO) as a model for a computer protocol architecture and as a framework for developing protocol standards The OSI model consists of seven layers namely:1 application, 2 presentation, 3 session, 4 transport, 5 network, 6 data link and 7 physical layer (Kurose & Ross, 2013).

Application layer provides access to the OSI environment for users while presentation layer provides independence to the application processes from differences in data representation (syntax). Session provides the control structure for communication between applications; establishes, manages, and terminates connections (sessions) between cooperating applications. Transport provides reliable, transparent transfer of data between end points; provides end-to-end error recovery and flow control.

Network provides upper layers with independence from the data transmission and switching technologies used to connect systems; responsible for establishing,

maintaining, and terminating connections. Data Link provides for the reliable transfer of information across the physical link; sends blocks (frames) with the necessary synchronization, error control, and flow control. Physical concerned with transmission of unstructured bit stream over physical medium; deals with the mechanical, electrical, functional, and procedural characteristics to access the physical medium(Kurose & Ross 2013; Bonaventure, 2010; Stalling, 2007).

Besides OSI model, Pavlou (2011) has identified and matched it to TCP/IP reference model. TCP/IP is based on five independent layers namely: 1. Physical/network interface layer 2. internet layer 3. transport layer and 4. application layer.

Application layer is the top most layer of four layer TCP/IP model. Application layer is present on the top of the Transport layer. It defines TCP/IP application protocols and how host programs interface with Transport layer services to use the network. This includes all the higher-level protocols like DNS (Domain Naming System), HTTP (Hypertext Transfer Protocol), Telnet, SSH, FTP (File Transfer Protocol), TFTP (Trivial File Transfer Protocol), SNMP (Simple Network Management Protocol), SMTP (Simple Mail Transfer Protocol) , DHCP (Dynamic Host Configuration Protocol) (Kurose & Ross 2013; Bonaventure, 2010; Stalling, 2007).

Transport Layer is the third layer of the four layer TCP/IP model. And its main purpose is to permit devices on the source and destination hosts to carry on a conversation. Transport layer defines the level of service and status of the connection used when transporting data. The main protocols included at this layer are TCP (Transmission Control Protocol) and UDP (User Datagram Protocol).

Internet Layer is the second layer of the four layer TCP/IP model. The position of Internet layer is between Network Access Layer and Transport layer. Internet layer pack data into data packets known as IP datagrams, which contain source and destination address (logical address or IP address) information that is used to forward the datagrams

between hosts and across networks. The Internet layer is also responsible for routing of IP datagrams.

Packet switching network depends upon a connectionless internetwork layer. This layer is known as Internet layer. Its job is to allow hosts to insert packets into any network and have them to deliver independently to the destination. At the destination side data packets may appear in a different order than they were sent. It is the job of the higher layers to rearrange them in order to deliver them to proper network applications operating at the Application layer. The main protocols included at Internet layer are IP (Internet Protocol), ICMP (Internet Control Message Protocol), ARP (Address Resolution Protocol), RARP (Reverse Address Resolution Protocol) and IGMP (Internet Group Management Protocol).

Network Access Layer is the first layer of the four layer TCP/IP model. It defines details of how data is physically sent through the network, including how bits are electrically or optically signaled by hardware devices that interface directly with a network medium, such as coaxial cable, optical fiber, or twisted pair copper wire.

Many users can be connected at the same time to a network of communication channels. Switching devices establish connections between nodes that need to communicate over a network. Principal techniques for switching include circuit switching and packet switching (Kurose & Ross 2013). In circuit-switched networks, the resources needed along a path (buffers, link bandwidth) to provide for communication between the end systems are reserved for the duration of the session. In packet-switched networks, these resources are not reserved; a session's messages use the resource on demand, and as a consequence, may have to wait (i.e., queue) for access to a communication link (Bonaventure, 2010).

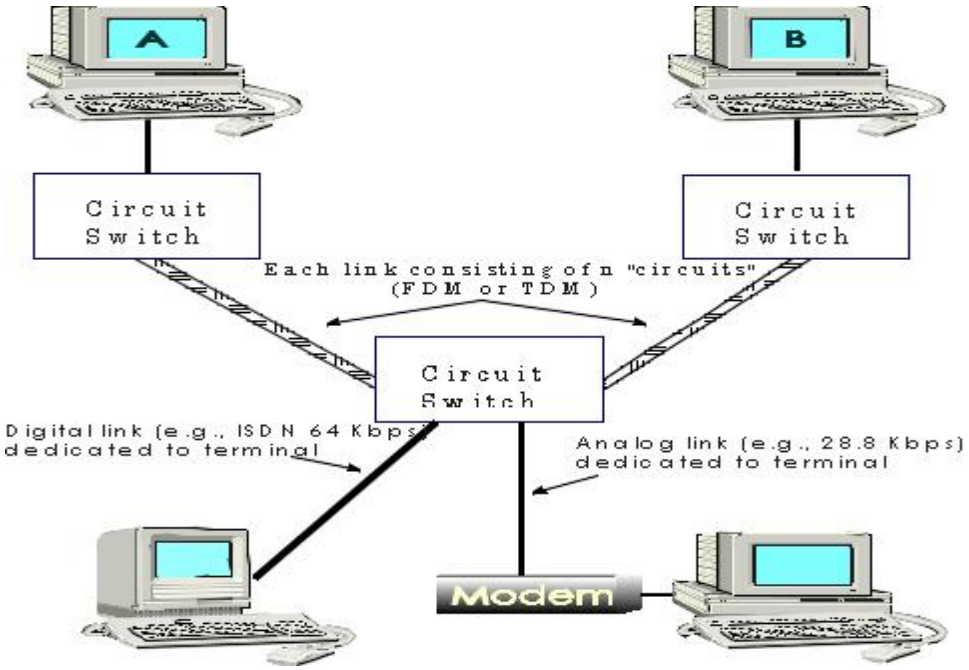


Figure 2.3 Circuit switched network. (Source: Kurose and Ross, 2013, p.30)

Figure 2.3 represents simple circuit-switched network consisting of three circuit switches interconnected with two links. Each link has n circuits; each end-to-end circuit over a link gets the fraction 1/n of the link's bandwidth for the duration of the circuit. The n circuits in a link can be either Time Division Multiplexing or Frequency Division Multiplexing circuits. A circuit in a link is implemented with either frequency division multiplexing (FDM) or time-division multiplexing (TDM). With FDM, the frequency spectrum of a link is shared among the connections established across the link (Bonaventure, 2010). Specifically, the link dedicates a frequency band to each connection for the duration of the connection. In telephone networks, this frequency band typically has a width of 4 kHz (Kurose & Ross, 2013). The width of the band is called the bandwidth. FM radio stations also use FDM to share microwave frequency spectrum (Stalling, 2007).

Packet switching has gained popularity in modern communication owing to its speed and its superior utilization of communication links when handling Burst and intermittent, traffic. Indeed, data transmission involves short bursts of activity by a computer or a terminal when the data are sent, followed by long periods when there is no transmission.

It offers flexibility in connecting to a network. It is used by most of the public data networks provided by value-added carriers(Kurose & Ross, 2007).

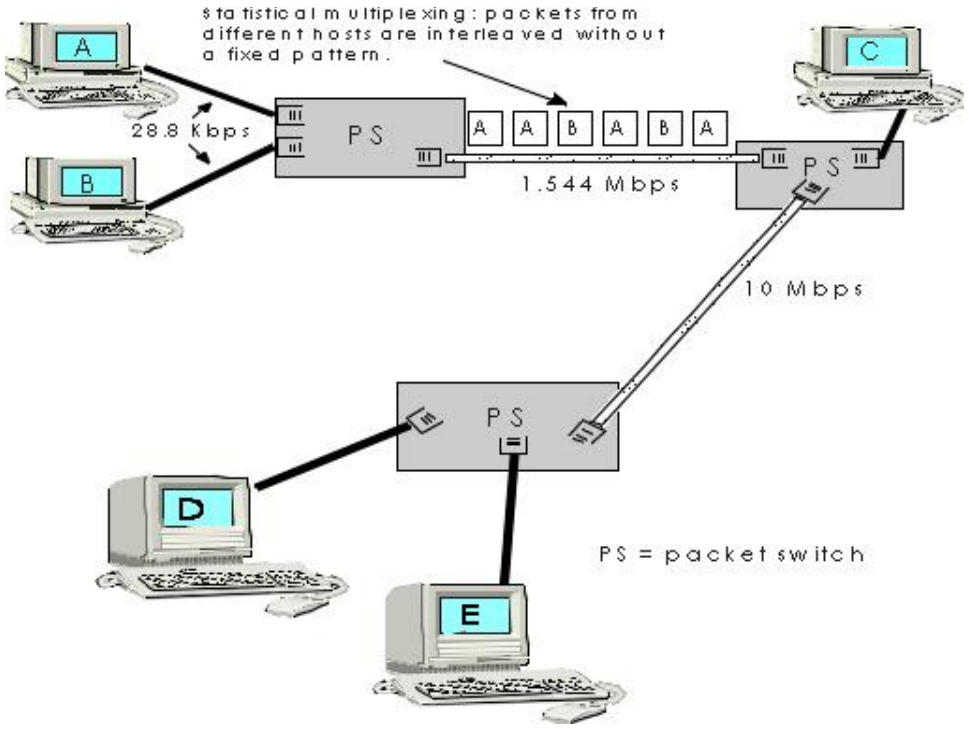


Figure 2.4: Packet Switching. (Source Kurose and Ross, 2013, p.33).

In modern packet-switched networks, the source breaks long messages into smaller packets. Between source and destination, each of these packets traverses communication links and packet switches (Bonaventure, 2010). Packets are transmitted over each communication link at a rate equal to the full transmission rate of the link (Kuross & Ross, 2013). Most packet switches use store and forward transmission at the inputs to the links. Store-and-forward transmission means that the switch must receive the entire packet before it can begin to transmit the first bit of the packet onto the outbound link.

Since most of internet applications have turned to packet switching and routing, protocols have been implemented to determine how packets are moved from source to destination a function usually performed by routers. Tanenbaum & Wetherall (2003) points out that routing packets involves passing them through intermediate nodes such as routers, bridges, gateways, firewalls, computers or switches for it to reach its destination Figure 2.5 gives a demonstration to understand the basic principal of routing process.

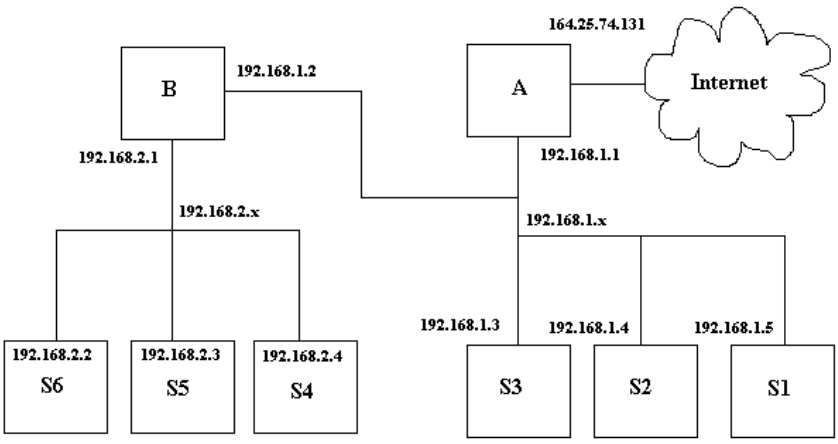


Figure 2.5: Typical network set up environment.

The boxes labeled A and B are routers in order for anyone on networks 192.168.2.x or 192.168.1.x to talk to any other network or internet. The boxes labeled S1 through S6 are stations which could be workstations or servers providing services like BOOTP, DHCP, DNS, HTTP, and/or file sharing such as NFS or Samba (Bengt, 2010). The gateways may also provide these services.

For example machine S6 in the figure 2.4 has IP Address of 192.168.2.2 which uniquely identifies it from other machines in the same network. It belongs to a Network of ID 192.168.2.0, subnetwork of Netmask 255.255.255.0 and gateway or router of ID 192.168.2.1.

This information is kept in a routing table and usually all computers that are networked have a routing table in one form or another. Pavlou (2011) defines a routing table as a simple set of rules that tell what will be done with network packets. Schmid (2007) provides a good example of a decision process. for example you are holding a letter. If it is addressed to you, you keep it, if it is addressed to someone in your town, you drop it in the local slot at the post office, but if it is addressed to someone out of town, you would drop it in the out of town slot. It is arranged from the most specific to the least specific. Therefore as you go down the table, more possibilities are covered that is netmask is 255.255.255.255 and the last is 0.0.0.0. There can be no doubt that the last

line is the default. The netmasks between the start and the end have a decreasing number of least significant bits set. The routing table for machine B, the gateway for the network 192.168.2.0 is as follows.

Table 2.1 Representation of a routing table

Destination	Gateway	Netmask	Ref	Use	Iface
192.168.2.1	*	255.255.255.255	0	0	eth0
192.168.1.2	*	255.255.255.255	0	0	eth1
192.168.2.0	192.168.2.1	255.255.255.0	0	0	eth0
192.168.2.0	*	255.255.255.0	0	0	eth0
192.168.1.0	192.168.1.2	255.255.255.0	0	0	eth1
192.168.1.0	*	255.255.255.0	0	0	eth1
127.0.0.0	*	255.0.0.0	0	0	Lo
Default	192.168.1.1	0.0.0.0	0	0	eth0

The Iface specifies the card where packets for this route will be sent. The address of eth1 is 192.168.1.2 and eth0 is 192.168.2.1. The NIC card addresses could have easily been switched. Line 1 provides for the eth0 address, while line 2 provides for the address of eth1. Lines 3 and 4 are the rules for traffic going from network 192.168.1.0 to network 192.168.2.0 which will be sent out on NIC eth0. Lines 5 and 6 are the rules for traffic going from network 192.168.2.0 to network 192.168.1.0 which will be sent out NIC eth1. Note the first value on lines 3 and 4 is 192.168.2.0 which the header indicates as the destination of the packet. The last line is the default line which specifies that any packet not on one of the networks 192.168.1.0 or 192.168.2.0 will be sent to the gateway 192.168.1.1. In case of overlapping or equal routes, elements such as prefix length, metrics and administrative distance are considered in order to decide which routes get installed into the routing table (Nelson, 2006).

The networks that maps IP address to end device are categorized under Host centric networks (Pavlou, 2011). Dolvara (2013) identifies a new approach to networking called Information centric networking. This approach handles packet routing and forwarding through two different methods, Name Resolution and Name-Based Routing. Name Resolution approach provides a means for client to search for Named Data Object (NDO) by name. It consists of two steps, the name to source locators mapping, and the forwarding of request message to the source. This two step approach needs additional entity called Name Resolution System (NRS) to provide the translation (Pavlou, 2011). The disadvantage of this approach is that NRS can be a point-of-failure, and as a consequence, many NDO registered on that NRS would be inaccessible. Another drawback is that the NRS requires a large storage to store NDO mapping. However, this approach guarantees the finding of the requested NDO since NRS already provides a pointer to the NDO source (Bengt 2010).

Name-Based Routing is a one step approach. NDO request is forwarded by content routers (CR), where CR locally decides the next hop of the NDO request based on NDO name. There are two types of routing models (Choi, 2010). The first is the Unstructured Routing, which is similar to the traditional IP routing with some modification. Therefore, it does not work well when there is NDO movement and when the number of copies of NDOs is large. The second model is the Structured Routing. It uses Distributed Hash Table (DHT) to provide a lookup and routing service. The advantage of this method is its ability to scale and a small overhead when changes occur. Name-Based Routing approach does not guarantee the discovery of the NDO since the resolving and forwarding process is done hop-by-hop (Dolvara, 2013).

Nelson (2006) has borrowed a Steiner mathematical algorithm for finding shortest path in a routing channel. The algorithm is formulated as follows:

Given a set P of n points, determine a set S of Steiner points such that the minimum spanning tree (MST) cost over $P \cup S$ is minimized. An optimal solution to this problem is referred to as a Steiner minimal over P, denoted as *SMT(P).* An edge in a tree *T* has cost equal to the distance between its endpoints, and the cost of *T* itself is the sum of its

edge costs, denoted cost(T). The wiring cost between a pair of pins (x1, y1) and (x2, y2) in a VLSI layout is typically modeled by the rectilinear distance (Nelson 2006):

$$dist((x_1, y_1), (x_2, y_2)) = (\Delta x) + (\Delta y) = |x_1 - x_2| + |y_1 - y_2| \quad (2.1)$$

This algorithm is of importance to this research as it provides an understanding of how the shortest route in a network set up is determined (Nelson, 2006).

While routing takes, large amount of content is transferred repeatedly (Dolvara, 2011). Most of the time, the content is retransmitted from the source to serve different requests coming across the network. Caching has been introduced as an effort to reduce the amount of repeated traffic in network setups. Network caching technique keeps frequently accessed information in a location close to the requester (Pavlou, 2011).

For example a Web cache stores Web pages and content on a storage device that is physically or logically closer to the user-closer and faster than a Web lookup. By reducing the amount of traffic on WAN links and on overburdened Web servers, caching provides significant benefits to ISPs, enterprise networks, and end users (Stolyar, 2010). Implementing caching technology localizes traffic patterns and addresses network traffic overload problems as the content is delivered to users at accelerated rates, WAN bandwidth usage is optimized and administrators can more easily monitor traffic. For example in figure 2.6 an ISP offering dedicated links to their clients has implemented distributed cache so that similar requested data can be retrieved from a cache making it possible to offer for services such as PPPoE(Point-to-Point Protocol over Ethernet) .

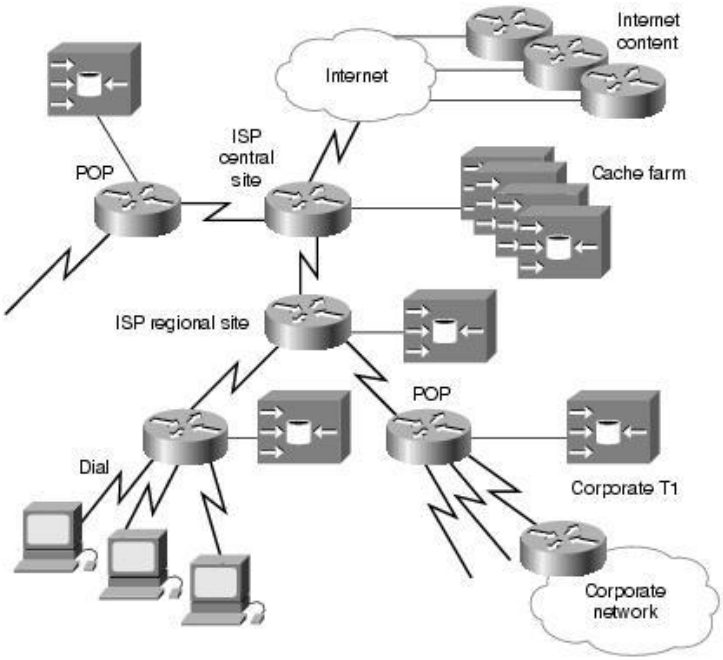


Figure 2.6: Distributed network cache. (Kurose & Ross, 2013, p.72).

Mironov (2010) classifies Caching as in-network caching where the caching is done within the networks and the edge caching where the end nodes store the cache. Furthermore, caching can also be divided into three levels based on granularity; object level (caching whole information object), chunk level(caching information chunks) and packet level(caching individual packets) (Paulo & Rute, 2010). Cached packets are given a hash value which uniquely identifies them a process called fingerprinting (pavlou ,2011; Schmidt , 2008).

Fingerprinting is an approach which is used to uniquely identify digital content in computer devices (Pavlou, 2011). A digital fingerprint is a unique representation of digital data in cache or storage media. This method enables a computer file to be encrypted such that it is computationally infeasible to obtain the same fingerprints from two files (Dolvara, 2010). This is an analogy which has been adopted from human being as each person has a unique fingerprint pattern (Bringmann, 2009; Shirley, 2014).

Message Digest Algorithm (MD5), Rabin and Secure Hashing Algorithm (SHA) are the major types of fingerprinting algorithms (Bringmann, 2009). Message Digest that is created when the algorithm is applied to a file consists of a unique String of 128 bits

(MD5), 160 bits (SHA) or 288 bits (MD5-SHA combined). Hexadecimal representation is utilized by fingerprinting mechanism (Minorov, 2005).

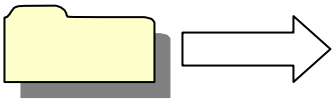


Figure 2.7: Fingerprinting mechanism (Source: Minorov, 2005)

Rabin fingerprints utilizes compounding and comes with a mathematical precise analysis for probability of collision namely, the probability of two strings r and s yielding the same w-bit fingerprint does not exceed

$$\max(|r|, |s|)\,|2^{w-1} \quad (2.2)$$

where |r| denotes the length of r in bits.

The algorithm requires the previous choice of a w-bit in internal "key", and this guarantee holds as long as the string r and s are chosen without the knowledge of the key(Bringmann, 2009). The advantage with rabin fingerprinting it takes short time to execute however sometimes it can be biased when it is used in chunking algorithms. It is also prone to security attacks as it is easy to discover the key and use it to modify files without changing their fingerprints (Rabin, 1999).

Figure 2.8: Rabin fingerprinting.(Source: Mironov, 2005, p.43)

In figure 2.8 a marker is chosen after every 4 bytes resulting into F(04), F(09), F(13) and F(17) sample for analysis. Usually rabin fingerprinting require multiplication and addition.

$$F = C_0 a^{k-1} + C_0 a^{k-2} + C_0 a^{k-3} + \cdots \ldots \ldots \ldots \ldots \ldots + C_{k-1} a^0 \quad (2.3)$$

The calculation of a new fingerprint from the old one just requires one addition, one subtraction and multiplication by "a"(Mironov, 2005).

SHA-1 fingerprinting utilizes use of compare–by-hash algorithm for calculations based on assumption that range from unproven to demonstrably wrong. The shortest lifetime and fast transition into obsolescence of cryptographic hashes makes them unsuitable for use in long lived systems. When hash collisions do occur they cause silent errors and bugs that are difficult to repair. SHA-1,SH-2 and SH-3 produces 160-bit (20 bytes) hash value usually rendered in hexadecimal number 40 digit long(Yang & Park, 2008).

Digital fingerprints are useful for verifying the integrity of file transmission. Fingerprinting has gained much use in cryptography where public key is used to encrypt cipher text and private keys to decrypt plain text for secure data transmission in networks.

Fingerprinting enables storage devices to identify duplicate files. Duplicate record elimantion theory developed by Dina Bitton is currently used for eliminating duplicate data file in database storage using a hash function and a bit array to determine whether two records are identical(Dina & David, 2003).

The algorithm finds distinct elements in a multiset

$$\{x_1 , x_2, \dots \dots \dots x_n\} \tag{2.4}$$

of not necessarily distinct elements. It is assumed that any two of these elements can be compared yielding

$$\{x_i > x_i \text{ and } x_i <= x_j\} \tag{2.5}$$

The x_i' s may be real numbers or alphanumeric strings that can be compared according to the lexicographic order, or they may be records with multiple alphanumeric fields, with one (or a subset of fields) used as the comparison key. The elements in the multiset (t) are duplicated according to some distribution

$$\{f_1 f_2, \dots \dots \dots f_m\} \tag{2.6}$$

That is, there are f_i elements with a "value" ~1, f_i elements with a value up to f_m elements with a value

v, and f_i = n. (2.7)

When n is large and the values are uniformly distributed, we may assume that

$$tf_i = f_i, \ldots\ldots = f_m = f \tag{2.8}$$

$$n = f * m$$

In this case, we define f as the "duplication factor" of the multiset. The theory will be relevant to this research study as the concept of duplicate entries will be applied to network packet byte redundancy.

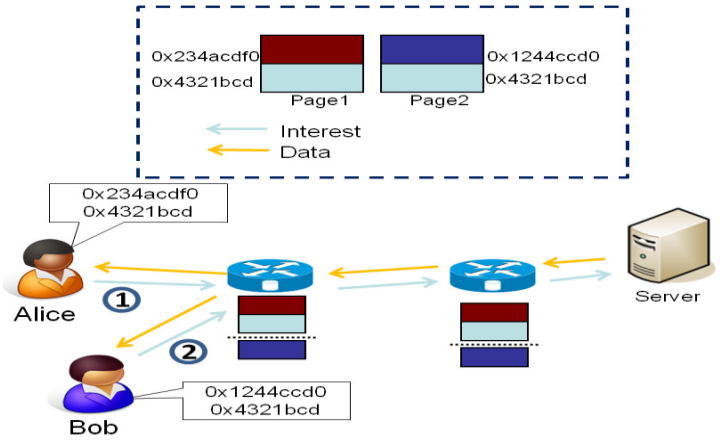


Figure 2.9: Redundancy bytes in information centric network packets (Source: Pavlou, 2011, p.23).

For example, two users Alice and Bob request two partially redundant documents. Page1 and page2, contain content names /bell_labs/page1.html and /bell_labs/page2.html composed by two 2,000 bytes chunks with fingerprints 0x234acdf0-0x4321bcd and 0x1244ccd0-0x4321bcd, respectively. But chunk with fingerprint 0x4321bcd is requested by both alice and bob making it to be cached twice

As data traverses the internet, its file type size and structure must be determined. Dina and David (2003) defines file as a collection of data, referred to by a specific name, residing on a storage device. Files are stored on hard disks, CDs, DVDs, flash drives, memory cards, network cache and so on. Broadly speaking, there are two types of files: text files and binary files. Text files contain only printable characters, that is, letters, numbers, spaces, tabs, and punctuation. Sometimes text files are called ASCII FILES (after the American Standard Code for Information Interchange). If you look inside a text file, what you see is readable by a human being (Bonaventure, 2011).

Binary files or binaries, on the other hand, contain non-textual data that makes sense only when read by a program. For example, if you were to look inside a music file, it would look like gibberish. A music file only makes sense when interpreted by a music program. Thus, all MP3 files (music files) are binary files. Other examples are executable programs, images, video files, word processing documents, spreadsheets, and databases. In terms of transmitting files from one place to another, a file can be transmitted as a "binary," meaning that the programs handling it don't attempt to look within it or change it, but just pass it along as a "chunk of 0s and 1s," the meaning of which is unknown to any network device(Schmidt, 2008).

2.2 Bandwidth Optimization Techniques

Munir (2014) notes that a communication link with high bandwidth is one that may be able to carry enough information to sustain the succession of data without jitter or buffering caused by latency. Mochizuki (2012) has defined bandwidth management as a general term given to collection of tools and techniques that institution can use to reduce network congestions.

Most network setups are faced with challenges in their use of networked information resources simply because the price of bandwidth is high hence researchers have proposed some of the methods to deal with this challenge (Schmidt, 2008; Koponen, 2007).

Bandwidth throttling is technique which is being used by network administrator by blocking some links on the network over a period of time in order to conserve bandwidth (Sellapan, 2014).

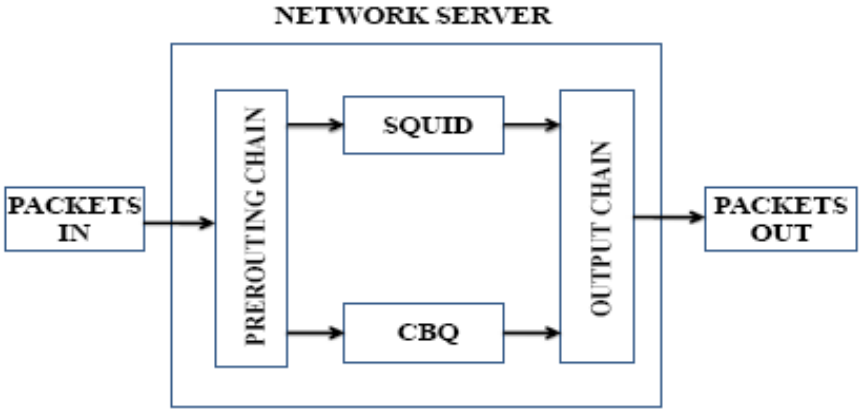


Figure 2.10: Bandwidth throttling. (Source: Munir, 2014, p.29).

Packets are inspected by pre-routing chain to determine those which will be allowed in first. SQUID fine tunes rate at which packets are transmitted while Class Based Queuing (CBQ) limits any packet that passes SQUID. OUTPUT CHAIN determines packets which will go out according to their priority. However bandwidth throttling sometimes can lead to service unavailability and denial of service.

Elimination of redundant data transfers has been deployed to reduce network load and improve network efficiency (Justine & Shaddi, 2013). Major techniques which have been deployed are middlebox optimizers, ENDRE and chunk level RE.

Middle boxes are placed at network points to detect and eliminate redundant packets (Anand, Sekar,& Akella, 2009). They improve bandwidth consumption and perceived latency between dedicated endpoints. Typically they are deployed in large enterprises both sending and receiving endpoints of communication. The devices then coordinate to cache and compress traffic that traverses the Internet.

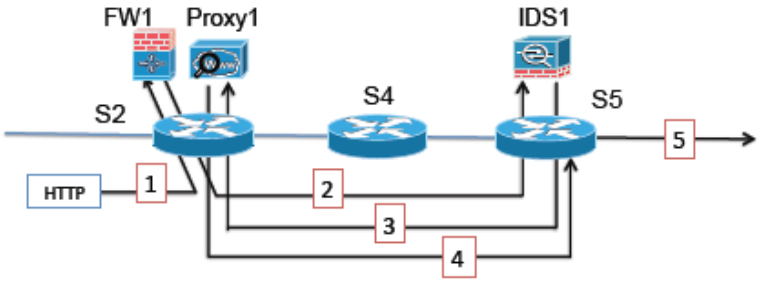


Figure 2.11: Middle box devices placed in between routers (Source: Munir, 2014, p.31).

Figure 2.11 is an example of a network setup with middle box devices such as firewall, proxy server IDS1, Wide area Network optimizer and routers s1, s2 and s3. Firewall and proxy server provide critical performance, security, and policy compliance functions while WAN optimizers provide data redundancy elimination mechanism. However Middle box redundancy elimination devices are expensive, complex to manage, and creates new failure modes for the networks that use them.

END-RE redundancy mechanism has been proposed in information centric networks as a software solution to middle box devices. Software which filters incoming and outgoing packets is placed at network endpoint.

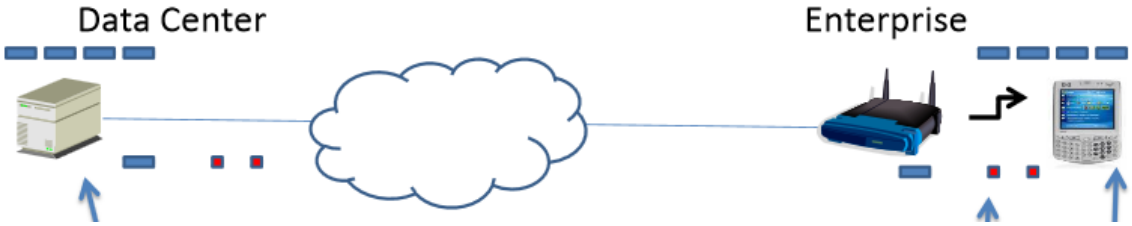


Figure 2.12: Network setup which has implemented ENDRE mechanism (Source: Halgren, 2012,p.17)

In figure Fig 2.12 software is installed at network center to filter incoming data from data center and outgoing data from enterprise. This mechanism uses Rabin Fingerprinting to chunk and analyze packets in transit. Rabin fingerprinting uses a

marker after every P bytes and then compares the sampled chunks to identify similarities.

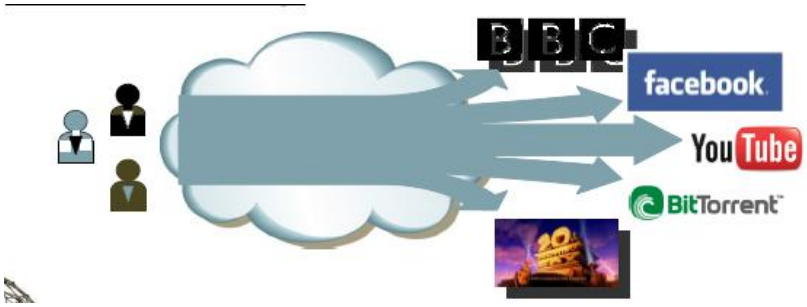
However rabin fingerprinting may result into low bandwidth saving as it may be biased at times. It has been proven that ENDRE may not be applicable for networks consisting of mobile devices which keep on changing positions and IP addresses making them difficult to capture redundant data transfers inside network setup (Munir, 2014).

Following Halgren (2012) study, effective management and optimization of bandwidth is critical as it will reduce running cost, reduce denial of service increase network efficiency. Moreover content passed over the internet is increasing every day making capacity development within the area of bandwidth management an essential element (Vaidya, 2011).

2.3 Information Centric model

This is a model that considers pieces of information as first-class entities of a networking architecture, rather than only indirectly identifying and manipulating them via a node hosting that information (Bengt, 2010). This paradigm shift is caused by the tremendous growth of information on the Internet and the increased demands for data access (Dolvara, 2013). Furthermore the current solution such as TCP/IP becomes inefficient and subjects to certain problems. For example, to search for a specific content, the piece of information must be mapped to a host, and then the DNS translates the host name to the location i.e. IP address. The two-step mapping incurs access overhead. IP address also binds to a location so it does not support node and content mobility. Security is another important issue since the security mechanism is tightly coupled to the host. Therefore, host becomes a target of many security threats. In addition, because the IP router is stateless and does not provide a caching capability, the same request would be made multiple times through the path, and cause an unnecessary bandwidth usage. This prompted the research into shifting Internet architecture from a host-centric to information centric (Pavlou, 2011).

a) Node centric design: sharing network resources



b)Information centric design: content access and distribution

Figure 2.13: Host centric versus information Centric Design. (Source: Pavlou , 2011,p.8)

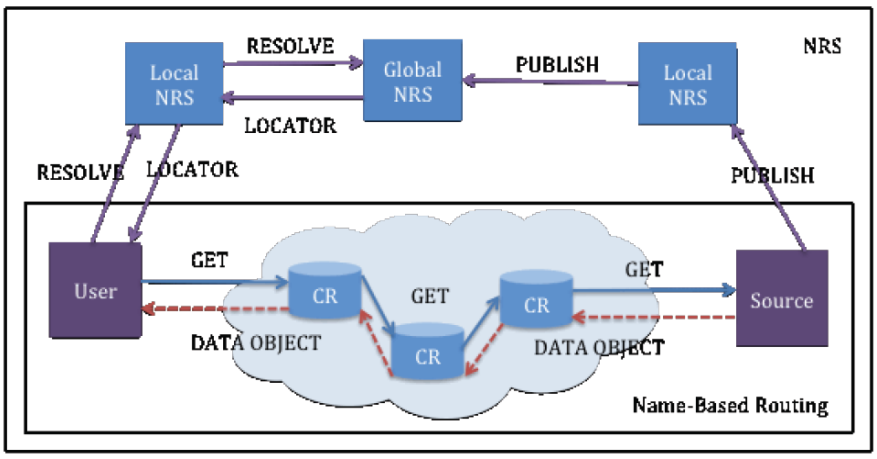


Figure 2.14 information centric network operations. (Source:Paulo Rute, 2010).

For information centric system, A source, who owns a copy of Named Data Object, sends a PUBLISH message to its local Name Resolution System. The local Name Resolution System then registers the Named Data Objects to the global Name Resolution System. A client sends a name resolution request to the local Name Resolution System.

If there is no match, a request is forwarded to global Name Resolution System, which returns an Named Data Object locator to the client. The client then requests for an Named Data Object by sending out GET message. The GET message is forwarded by the underlying network to the source. The data response is then route back from the source to client using the same path. In Named-Base Routing approach, a client sends a GET message directly to the content routers . Named Data Object information will forward the request to the Named Data Object source(Paulo & Rute, 2010).

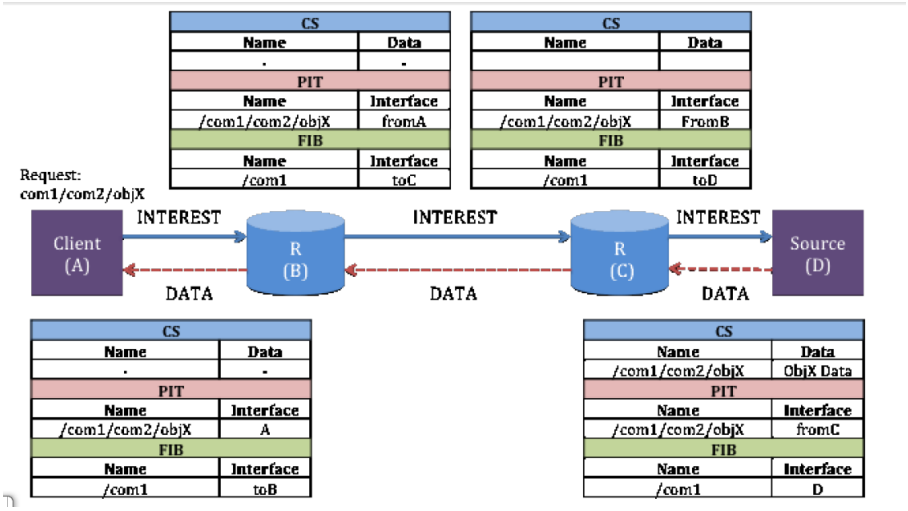


Figure 2.15. Information centric routing table. (Source: Dolvara, 2013, p.5)

Information unit in information centric network is generally called Named Data Object which can be any kind of digital content such as a video, an image, a document, a webpage or it can represent a real world object. It consists of a location-independent identifier , data, and possibly a metadata, which describes an named data object (Dolvara, 2013).

Information centric networks handles named data object packet routing and forwarding through two different methods, Name Resolution and Name-Based Routing object (Dolvara, 2013; Pavlou, 2011). Name Resolution approach provides a means for client to search for NDO by name. It consists of two steps, the name to source locators mapping, and the forwarding of request message to the source (Choi, 2010). This two step

approach needs additional entity called Name Resolution System to provide the translation. This approach guarantees the finding of the requested NDO since NRS already provides a pointer to the NDO source.

Name-Based Routing is a one step approach. NDO request is forwarded by content routers (CR), where CR locally decides the next hop of the NDO request based on NDO name. There are two types of routing models (Choi, 2010). The first is the Unstructured Routing, which is similar to the traditional IP routing with some modification making it not work well when there is NDO movement and when the number of copies of NDOs is large. The second model is the Structured Routing. It uses Distributed Hash Table to provide a lookup and routing service. The pros of this method are its ability to scale and a small overhead when changes occur. Name-Based Routing approach does not guarantee the discovery of the data named object since the resolving and forwarding process is done hop-by-hop (Dolvara, 2013).

ICN offers caching service to improve the performance of NDO access of the subsequent requests. Multiple copies of NDOs can be distributed across the networks. It can be stored locally on a node's cache or can be a shared on a network cache. Caching can also be classified into in-network caching where the caching is done within the networks, i.e., on the content routers, and the edge caching in which the end nodes store the cache. (Paulo & Rute, 2010).

2.4 Network Throughput

It is important to measure maximum data throughput of a communication link. One way to measure this is to transfer large files from one system to another and measure the time required to complete the transfer copy of the file, then dividing the file size by the time to get the throughput in megabits, kilobytes or bits per second(Schmdt, 2008). Although this measurement holds, studies reveal aspects such as network physical setup, traffic pattern and network stack implementation and configuration impact on the throughput (Neilson, 2006; Brinkmann 2009).

Beingt (2010) argues out that the physical aspects of the network need to be well understood. The physical connections between client, server, routers, switch, network cards and other network devices have an impact on network performance. The size of packets are transmitted through the network matters either packets that are fragmented or consolidated along the transmission path.

Traffic Pattern is another issue impacting networking throughput. Even traffic generated by the same application can be different depending on the groups that utilize the application and time of day. For example, an application may generate a steady stream of small packets when used by individual sales representatives, but the traffic may become bursty and consist of large packets when the area managers generate activity reports. bengt(2010) notes issues to be aware of when considering traffic patterns such as Frequency of the transactions and whether packets come in bursts; Size of the data packets; Sensitivity to data loss; for example, a multimedia streaming application using UDP may still present acceptable media quality to the user even when the data loss is as high as a few percent; Traffic directiveness as most of the time, network traffic is substantially asymmetric, with a lot more data transmitted downstream (from the server to the client) than upstream(Kurose & Ross, 2013).

Network protocol stack implementation in the operating system and application performance in processing network transactions often impacts overall network performance. With new network cards and switches now reaching 10Gbps, the bottleneck in processing network traffic often lies with available CPU cycles and system memory, whether it is for processing the transaction on the application level or for executing the operating system's TCP/IP stack (Stalling, 2007). Stalling (2007) suggested a formula to calculate network throughput(T).

$$T = \frac{RWIN}{RTT}, \qquad (2.9)$$

where $RWIN$ is the receive window size and RTT is the Round-Trip Time. For example given TCP Window of 65,535 bytes and RTT of 0.220s,

$$T \leq \frac{65635\ bytes}{0.220s} = \frac{297886.36\ bytes}{sec} = 2.383 megabytes/sec$$

2.5 Performance Evaluation Conceptual Framework

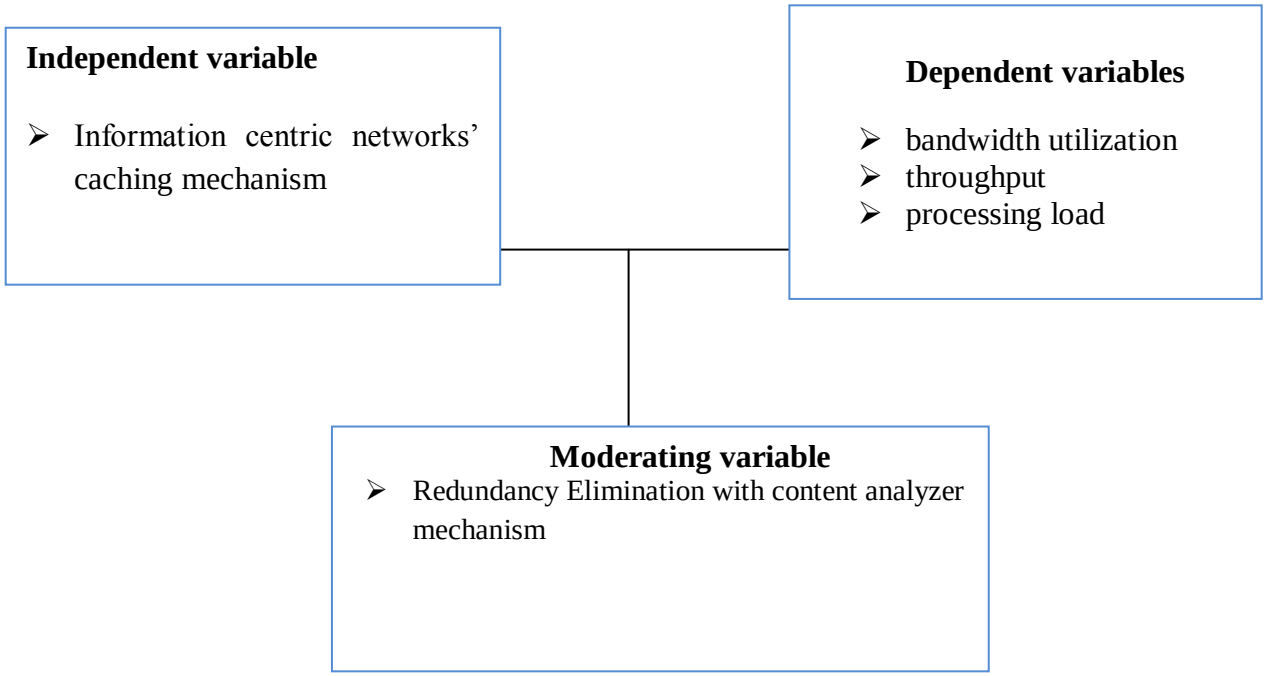


Figure 2.16: Conceptual framework

Figure 2.16 represents a performance evaluation conceptual framework for our enhanced model. Information centric networks designs are utilized in most of today organization setups since they have more features than Host Centric designs. Among this features they utilize the concept of caching to store frequently accessed information in the local content store instead of requesting them again from servers.

However, information centric networks need to manage bandwidth carefully in order to avoid problems such as denial of service or stuttering experiences. Power consumption also needs to be utilized well since power is a critical resource. The number of packets delivered at the destination at a given time need to be increased to increase network performance.

This research introduced Redundancy elimination mechanism to eliminate redundant chunks in transit and optimize bandwidth. It introduced a content analyzer to inspect data and make a decision whether to chunk (text files) or bypass (binary files) depending on the type of the file captured.

The First Stage, data inspection: using the content analyzer The Second Stage, RE Bypass: If binary flow skip chunking process transfer it to downstream router update the indexes and store data If content analyzer has identified non- binary files:

Step one is to divide data stream into small chunks (Brinkmann, 2009). Step two is to calculate hash function for each chunk using SHA-3 fingerprinting algorithm. Step three is to detect duplicate content by comparing hash results with already stored index. Step four is to update indexes and store data.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Research Design

This study employed experimental research design to measure efficiency of information centric model designed. It was found to be appropriate in our study since it is concerned with deliberately changing of one or more process variables in order to observe the effect the changes have on one or more response variables. It was also found to be an efficient procedure for planning experiments so that the data obtained can be analyzed to yield valid and objective conclusions (Nassiuma 2000). In our study packet samples were captured from different network setups their headers and contents analyzed to find percentage of repetitive contents.

3.2 Sample Size and Sampling Design

An important aspect of designing an experiment is to know how many observations are needed to make conclusions of sufficient accuracy and with sufficient confidence. This research used small, medium and large enterprises network environment that had proxy server, content routers, switches and gateways. We classified the enterprises as small, medium or large based on the number of internal host IP addresses they accommodated (less than 50, 50-100, and 100-250, respectively) in the entire trace at each of these sites. While this classification was somewhat arbitrary, we used this division to study if the benefits depended on the size of an enterprise. Note that the total amount of traffic in each trace was approximately correlated to the number of host IP addresses, though there was a large amount of variation from day to day. Typical incoming traffic numbers for small enterprises varied from 0.3-10GB/day, for medium enterprises from 2-12GB/day and for large enterprises from 7-50GB/day. The access link capacities at these sites varied from a few Mbps to several tens of Mbps. The total size of traffic we captured (including inbound/ outbound traffic and headers) was approximately 3.5TB.

Packet samples were captured between 1.00 am and 6.30 pm since it was found that this is the time networks are utilized by users.

Table 3.1: Network sites used for the study

Trace Site	Unique Client IPs	Dates(total days)	Size(TB)
Small enterprise	27-37	10/11/2014- 17/11/2014(7 days)	0.3
Medium enterprise	61-90	16/11/2014-22/11/2014(6 days)	1.1
Large enterprise	101-210	05/12/2014-14/12/2014(9 days)	2

Table 3.1 represents content stores of sites which reflected the traffic typically observed at the access points. The stores consisted of YouTube video files, music files, images, and executable files; totaling to 3.4 TB.

3.3 Experimental Setup

It is important to understand the set up environment where the research is carried out in order to make sure that the effects observed when manipulating the experimental setup are not just random effects due to chance. Our set up consists of the transmitter that is used to send data on the network. It also contains a receiver that accepts the transmitted information. The network monitoring module is connected to this network so as to capture and analyze the transmitted data, determine its source and destination as shown in figure 3.1. The experiment was set in a manner it could accommodate both wired (CAT5E Ethernet cable) and wireless devices (access point 802.11g enabled). All setups were connected to internet service provider via a router.

Figure 3.1: Experimental setup

3.4 Data Collection Methods

The research relied on data from both real traces and secondary data from other researchers which could be vital in our research. Full packet traces were collected in LAN access links for six busy sites with high uploads and downloads. We used wireshark packet traffic analyzer module(Laura, 2014) to collect data that included network parameters such as maximum throughput, source and destination address, the type of protocol implemented, the time to live of packets, bandwidth and bit rates, Input Output graph, frame sizes , packet content and traffic data types. The collection consisted of snapshot recorded morning and evening hours from 10/11/2014 to December 14th, 2014.

3.5 Data analysis and presentation

It was found appropriate to inspect data streaming particularly http data by use of a parser program. A parser is a program that receives input in the form of sequential source program instructions, interactive online commands, markup tags, or some other defined interface and breaks them up into parts that can give useful information about the input data. This parser could be important as it could enable us to add a bypass for binary data in our chunking algorithm.

Using PHP we designed HTTP header parser which inspected the initial replies arriving from the server, recorded the file type field in the response message and stored its value in memory. The parser developed was platform independent since it could execute in both windows and linux environment. We used the *file_get_contents()* function to read file into a string while the *filesize()* function was used to return the size of the specified file. On the other hand The *get_mime_type ()* function returned the file type of streamed data

Table 3.2: Code illustration of content analyzer

```
<?php
function get_mime_type($file)
{
      // our list of mime types
       $mime_types = array(
            "pdf"=>"application/pdf"
            ,"exe"=>"application/octet-stream"
            ,"zip"=>"application/zip"
            ,"docx"=>"application/msword"
            ,"doc"=>"application/msword"
            ,"xls"=>"application/vnd.ms-excel"
            ,"ppt"=>"application/vnd.ms-powerpoint"
            ,"gif"=>"image/gif"
            ,"png"=>"image/png"
            ,"jpeg"=>"image/jpg"
            ,"jpg"=>"image/jpg"
            ,"mp3"=>"audio/mpeg"
            ,"wav"=>"audio/x-wav"
            ,"mpeg"=>"video/mpeg"
            ,"mpg"=>"video/mpeg"
            ,"mpe"=>"video/mpeg"
            ,"mov"=>"video/quicktime"
            ,"avi"=>"video/x-msvideo"
            ,"3gp"=>"video/3gpp"
            ,"css"=>"text/css"
            ,"jsc"=>"application/javascript"
            ,"js"=>"application/javascript"
```

```
                ,"php"=>"text/html"
                ,"htm"=>"text/html"
                ,"html"=>"text/html"
            );
            $extension = strtolower(end(explode('.',$file)));
            return $mime_types[$extension];
    }    ?>
<?php

// simple calls to fetch mime type of a file

echo get_mime_type('jellyfish.jpeg'); // displays "application/msword"

echo get_mime_type(test.php'); // displays " text/html "

echo get_mime_type('Harry_Potter.mov'); // displays "video/quicktime"

?>
```

The collected data was analyzed and presented using descriptive statistics such as frequencies, percentages, flow charts and graphs. The data traces were analyzed to find out the percentage of different file types they represent. The study found out that 97% of total volume constituted binary files while 3% represented text files. This prompted us further to present a table to compare percentage of redundant contents in these file types that is in text, sound, movies and images. Graph to compare bandwidth saving of different bandwidth optimization techniques with our enhanced bandwidth optimization model was shown. Tabular and Graphical representation of costs of bandwidth before and after redundancy elimination across different ISP was presented. It was also vital to find out network power consumption rates of different bandwidth optimization mechanism. Graphical presentation comparing overall redundancy elimination to

different packet chunk sizes was presented. Finally a flow chart representing our enhanced bandwidth optimization model was presented.

CHAPTER FOUR
RESULTS AND DISCUSSION

4.0 Introduction

This chapter gives an explanation for the results that were obtained from the field study that was carried out. The probable explanation of the observed results and their analysis forms the basis of this chapter. The chapter is divided into the following key areas: Enhanced information centric model ,results and discussion from total volume of data captured; evaluation of percentage of bandwidth saving of different bandwidth management mechanism with our enhanced bandwidth optimization algorithm, comparison of processing load of different redundancy elimination algorithm with our enhanced Redundancy Elimination algorithm; comparison of throughput of current Redundancy elimination algorithms with our enhanced redundancy elimination model; discussion of the benefits of the enhanced redundancy elimination model.

4.1. Enhanced Information Centric model

It is evident that the price to pay for detecting redundancy in low binary files is high prompting the research to utilize this observation to increase the performance of existing RE algorithms, such as the ones presented by Munir (2014) and Diego (2012).

Our algorithm utilizes the fact that the probability of binary files partial intersection is minimal therefore utilizes this to reduce network processing load. It is a two way selection circuit evaluation process. In figure 4.1, a variable that analyzes the content to determine file type is introduced before the chunking process is done.

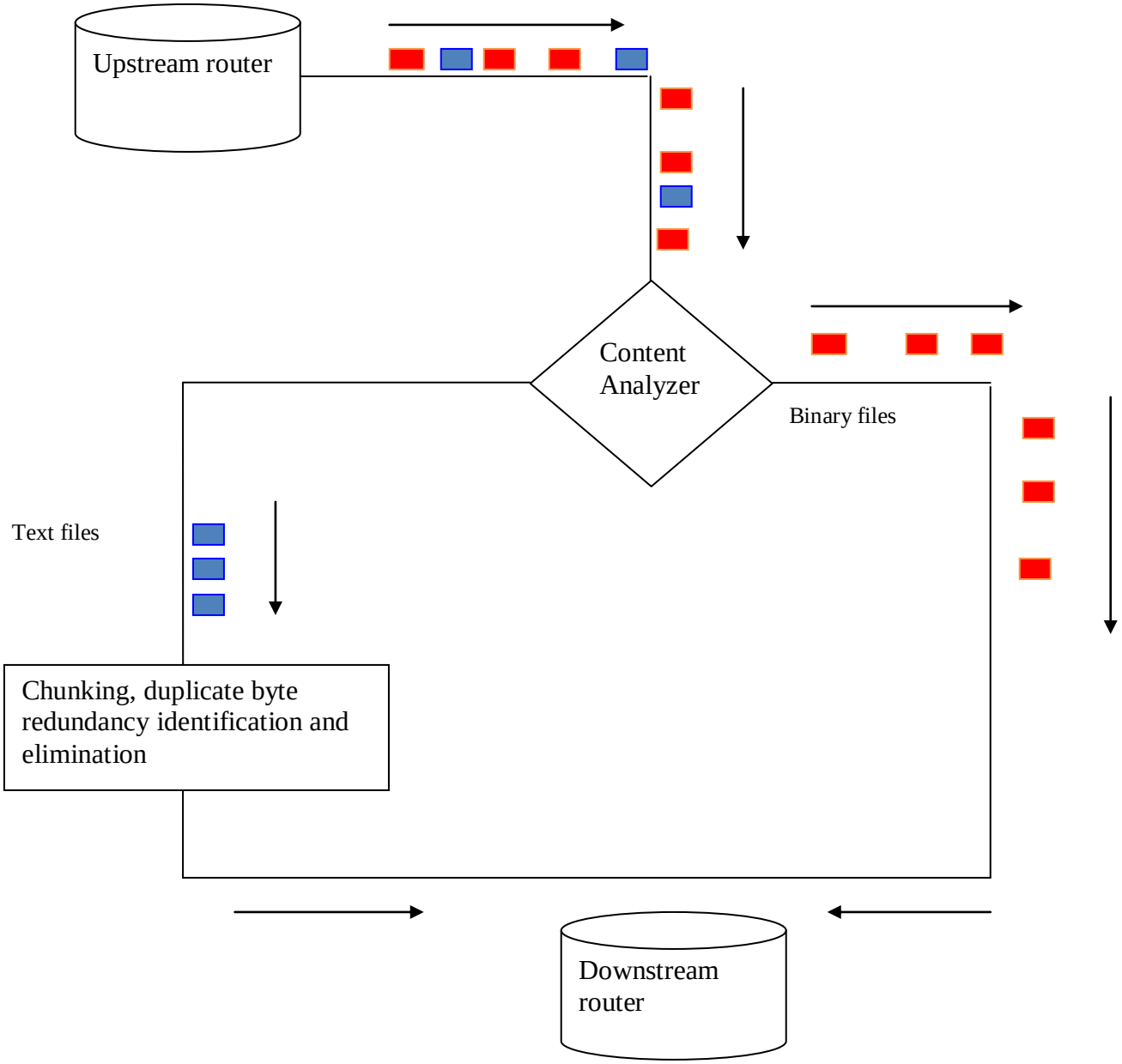


Figure 4.1: UML presentation of the research Model

The research algorithm has two major stages:

The First Stage, data inspection: it inspects each packet going through the RE engine for signatures that indicate the content type of the upcoming flow. This is applied to both HTTP,P2P or flash stream content. The HTTP OK response from the server is tracked and parsed for “mime-type” header field. This field typically indicates to the client

software the type of the payload contained in the response flow. The mimetypes that we used for our algorithm are binary files(“audio”, “video”,iamges ISO files and executable files) and text files(word,html,pdf) . When any of these binary mime-types are found in an HTTP OK response, that flow is marked as binary flow and is treated differently than other TCP flows. However, persistent HTTP connections pose a different challenge as they transfer multiple contents over the same TCP flow. To address this, the Content Detection stage also parses the HTTP OK packet for the “content-length” header field. This field indicates the length of the content that will be transferred before the next GET request is processed.

The Second Stage, RE Bypass: This step of the algorithm utilizes the information from the first stage to reduce the processing load, maintenance, and cache lookup time at the network element. This step is basically a modified version of a typical RE algorithm, such as CombiHeader , which bypasses the packets from the flow marked as binary-flow. This simple choice of bypassing ensures that we do not spend valuable processing power for chunking traffic which will most probably not match with any later chunks. The completely matched binary files, packet level redundancy aware algorithm. With this simple modification in RE algorithms operation, not only huge processing power is saved, but also the practical throughput increases due to less delay on the network element. If content analyzer has identified non- binary files fingerprinting takes place using four stages.

Step one is to divide data stream into small chunks using combination of static, content-defined and file-based chunking approaches (Brinkmann, 2009). Step two is to calculate hash function for each chunk using SHA-3 fingerprinting algorithm to calculate a Hash Value of each of the substring of size *w,* by use of the formula 4.1*:*

$$fmoden = c$$

Where (4.1)

$$0 \leq c < n$$

and f is fingerprint and *n* is expected length.

Step three is to detect duplicate content by comparing hash results with already stored index. Step four is to update indexes and store data

The research model in figure 4.1 can also be represented by use of algorithm. Our algorithm introduced a variable (***Binary-flow***) to solve the problem as follows:

Table 4.1. Enhanced Redundancy Elimination algorithm

While Steiner algorithm applies do

WHILE packet received do
If GET Request then
Set ***binary-flow***←0
Set content-length←0
Else if OK Response then
If MIME-Type is in[Video ,audio, application] then
Set ***binary-flow***←1 for the flow
Set content-length←Content length field
End if
Else
Find the flow of the packet
If content-length >0 and ***binary-flow***=1 then
Pass packet without chunking or caching
Else
If content-length=0 and ***binary-flow***=1 then
Binary-flow←0
End if
mask←0x0008A3110583080{48 Byte window; 8KB chunks}
longval←0{has to be 64 bits}
for all byte€ stream do

```
shift left longval by 1 bit {1sb←0; drop msb}
longvalue←longvalue bitwise-xor byte
if processed at least 48 bytes and(longval bitwise- and mask)== mask then
found an anchor
end if
end for
end if
end while
```

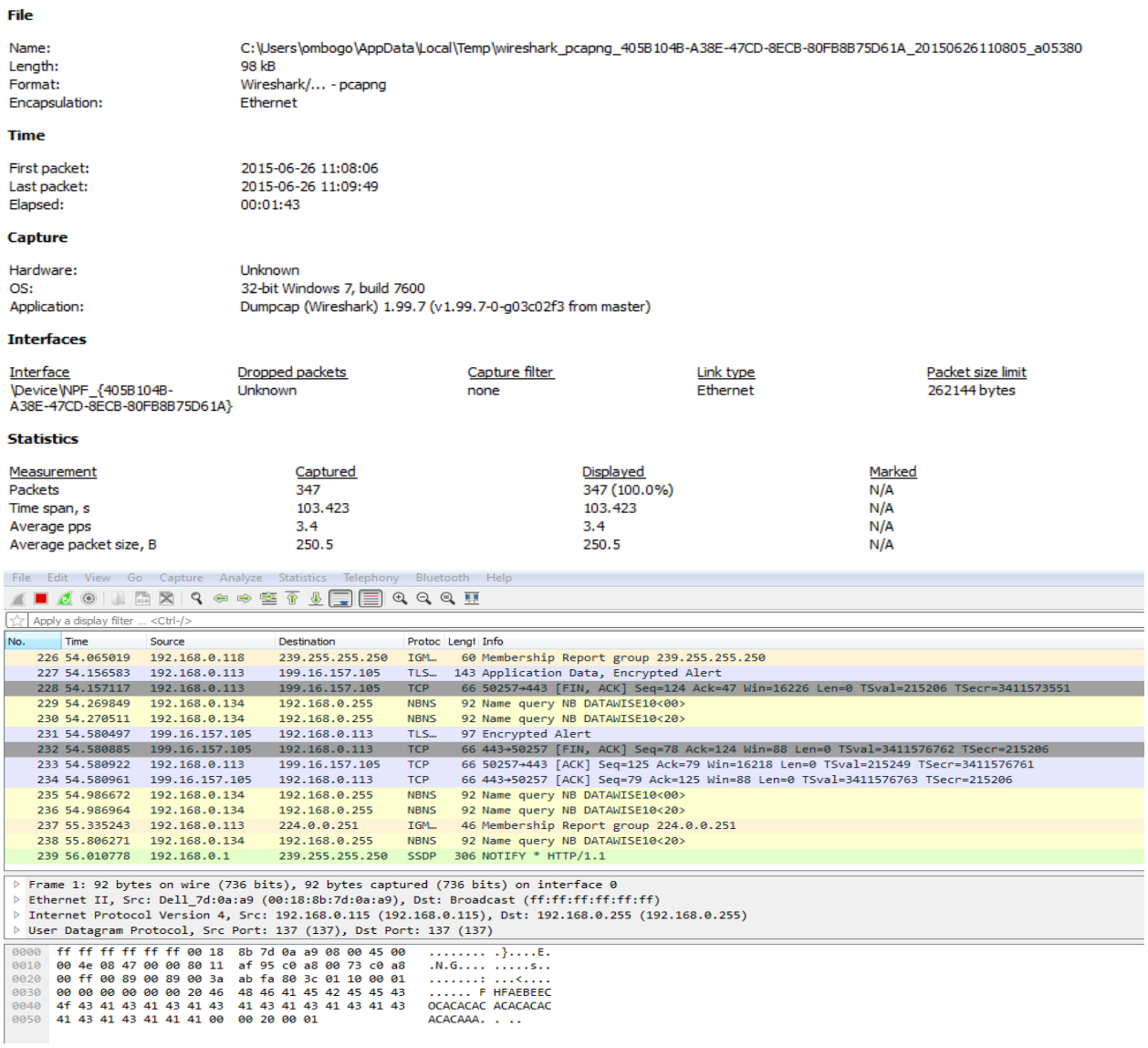
4.2 Content mix and order of files captured in the experiment

With the increasing and speed of internet speed, users are increasingly downloading and uploading content. This prompted us to study and analyze the sample size to determine percentage of file types streaming through the internet. It also prompted us to do a comparison between duplicated content and associated file type. We designed HTTP header parser as shown in table 3.2 which inspects the initial replies arriving from the server, the parser looks for the file type field in the response message and stores its value.

Table 4.2 Showing sizes of different files captured from content store

File NO.	File Name	File Size(MB)
1	00Illusion.mp3	102
2	Brandy.avi	67003
3	Chiquitita.mp3	510
4	Collection of text files	101
5	5.jpg	301
6	collection of text files	241
7	Collection of text files	6182
8	11.mp3	654
9	15.mp3	815
10	AVSEQ03.DAT	8189
11	Collection of text files	557
	TOTAL	**84655**

The research collected different file types from the content store and found out that 97% of total volume constituted binary files such as audio, video and pictures. 3% of file the volume constituted text files such as pdf, Microsoft office word, PowerPoint and html files. This was made possible by use of Wireshark packet analyzer and http parser tool



File

Name:	C:\Users\ombogo\AppData\Local\Temp\wireshark_pcapng_405B104B-A38E-47CD-8ECB-80FB8B75D61A_20150626110805_a05380
Length:	98 kB
Format:	Wireshark/... - pcapng
Encapsulation:	Ethernet

Time

First packet:	2015-06-26 11:08:06
Last packet:	2015-06-26 11:09:49
Elapsed:	00:01:43

Capture

Hardware:	Unknown
OS:	32-bit Windows 7, build 7600
Application:	Dumpcap (Wireshark) 1.99.7 (v1.99.7-0-g03c02f3 from master)

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
\Device\NPF_{405B104B-A38E-47CD-8ECB-80FB8B75D61A}	Unknown	none	Ethernet	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	347	347 (100.0%)	N/A
Time span, s	103.423	103.423	N/A
Average pps	3.4	3.4	N/A
Average packet size, B	250.5	250.5	N/A

File Edit View Go Capture Analyze Statistics Telephony Bluetooth Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protoc	Lengt	Info
226	54.065019	192.168.0.118	239.255.255.250	IGM…	60	Membership Report group 239.255.255.250
227	54.156583	192.168.0.113	199.16.157.105	TLS…	143	Application Data, Encrypted Alert
228	54.157117	192.168.0.113	199.16.157.105	TCP	66	50257→443 [FIN, ACK] Seq=124 Ack=47 Win=16226 Len=0 TSval=215206 TSecr=3411573551
229	54.269849	192.168.0.134	192.168.0.255	NBNS	92	Name query NB DATAWISE10<00>
230	54.270511	192.168.0.134	192.168.0.255	NBNS	92	Name query NB DATAWISE10<20>
231	54.580497	199.16.157.105	192.168.0.113	TLS…	97	Encrypted Alert
232	54.580885	199.16.157.105	192.168.0.113	TCP	66	443→50257 [FIN, ACK] Seq=78 Ack=124 Win=88 Len=0 TSval=3411576762 TSecr=215206
233	54.580922	192.168.0.113	199.16.157.105	TCP	66	50257→443 [ACK] Seq=125 Ack=79 Win=16218 Len=0 TSval=215249 TSecr=3411576761
234	54.580961	199.16.157.105	192.168.0.113	TCP	66	443→50257 [ACK] Seq=79 Ack=125 Win=88 Len=0 TSval=3411576763 TSecr=215206
235	54.986672	192.168.0.134	192.168.0.255	NBNS	92	Name query NB DATAWISE10<00>
236	54.986964	192.168.0.134	192.168.0.255	NBNS	92	Name query NB DATAWISE10<20>
237	55.335243	192.168.0.113	224.0.0.251	IGM…	46	Membership Report group 224.0.0.251
238	55.806271	192.168.0.134	192.168.0.255	NBNS	92	Name query NB DATAWISE10<20>
239	56.010778	192.168.0.1	239.255.255.250	SSDP	306	NOTIFY * HTTP/1.1

```
▷ Frame 1: 92 bytes on wire (736 bits), 92 bytes captured (736 bits) on interface 0
▷ Ethernet II, Src: Dell_7d:0a:a9 (00:18:8b:7d:0a:a9), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
▷ Internet Protocol Version 4, Src: 192.168.0.115 (192.168.0.115), Dst: 192.168.0.255 (192.168.0.255)
▷ User Datagram Protocol, Src Port: 137 (137), Dst Port: 137 (137)
```

```
0000  ff ff ff ff ff ff 00 18  8b 7d 0a a9 08 00 45 00   ........ .}....E.
0010  00 4e 08 47 00 00 80 11  af 95 c0 a8 00 73 c0 a8   .N.G.... .....s..
0020  00 ff 00 89 00 89 00 3a  ab fa 80 3c 01 10 00 01   .......: ...<....
0030  00 00 00 00 00 00 20 46  48 46 41 45 42 45 45 43   ...... F HFAEBEEC
0040  4f 43 41 43 41 43 41 43  41 43 41 43 41 43 41 43   OCACACAC ACACACAC
0050  41 43 41 43 41 41 41 00  00 20 00 01               ACACAAA. . ..
```

Figure 4.2: Sample of captured packets

figure 4.2, shows sample packets in transit ,statistical aspects such as number of packet transmitted, time taken to transmit and comparison of packet contents.

4.3 Evaluation of bandwidth management techniques

The key to making most of network links efficient and reliable lies in grooming and reducing the traffic that travels across them, and in avoiding as many potential sources of delay as possible. Therefore it is important to evaluate the performance of each bandwidth techniques. Since each bandwidth management technique had its own way of

testing bandwidth optimization we had to set their experiment separately and record their findings as shown in section 4.3.1,4.3.2,4.3.3 and 4.3.4. This research summarized the findings from each bandwidth management technique and produced the graph in figure 4.3.

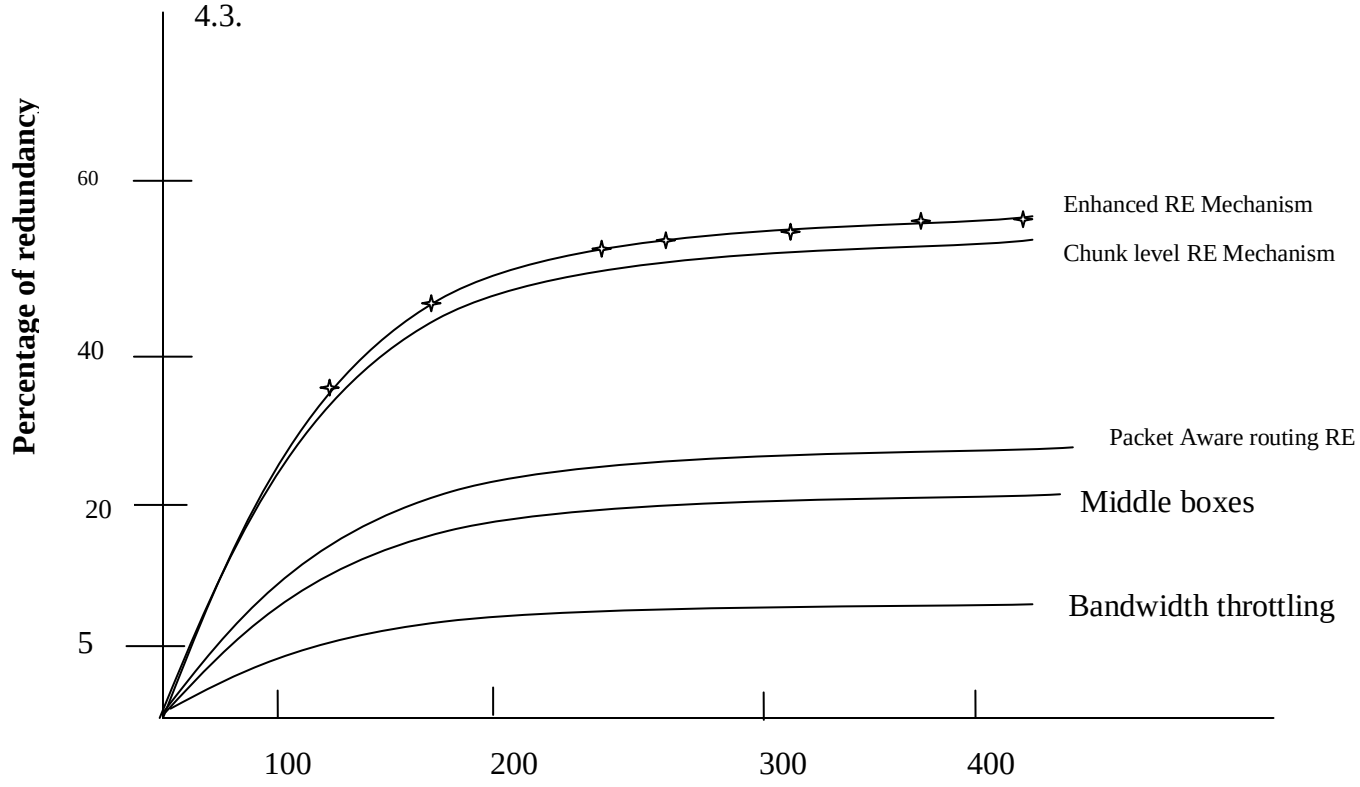


Figure 4.3: Graph comparing percentage of redundancy elimination produced by

F

bandwidth management techniques

due to the probability that part of the blocked portion contains redundant data. Middle boxes achieve 17% redundancy elimination since they are only placed at the end of network points. The result is also influenced by the fact that middle boxes track redundant traffic between devices connected to them and cannot track mobile or devices using wireless network. With packet routing redundancy aware routing we were only able to achieve 21% redundancy elimination better improvement as compared to bandwidth throttling. But with byte level without bypass we were able to achieve 43% redundancy elimination which was as a result of chunking individual packets. Byte level

redundancy elimination with bypass mechanism however produced high level of redundancy elimination (46%) and was optimal in terms of processing load making our algorithm to be the best in bandwidth optimization as compared to the rest.

4.3.1 Bandwidth throttling

This was the first experiment that we carried out to test the percentage of bandwidth saving that it could produce. We applied a proxy server to filter in and outbound messages. Our goal was to block some sites when bandwidth is exhausted specifically to heavy bandwidth usage sites such as video downloading links.. Among the test was to check if the quality of service has improved or not.

In doing this, we connected workstations to our Local Area Networks. After the connection was established, we tried to download an "iso" file from www.ubuntu.com/download and check the speed. The downloading started automatically and the speed was within the specified rate i.e. 512Kbps. files with ".iso" extension are members of bad_ extensions acl on the squid delay pool configuration.

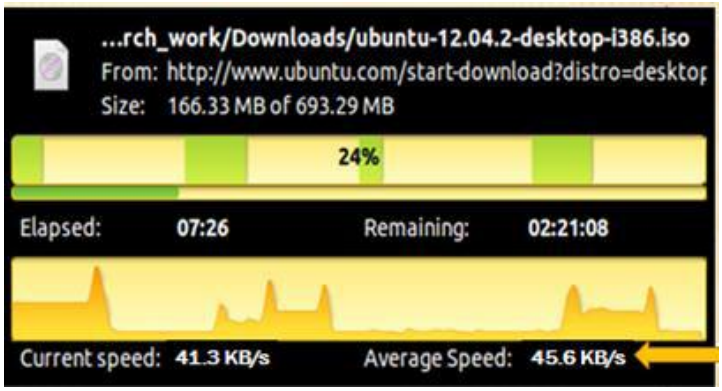


Figure 4.4: downloading speed before applying bandwidth throttling.

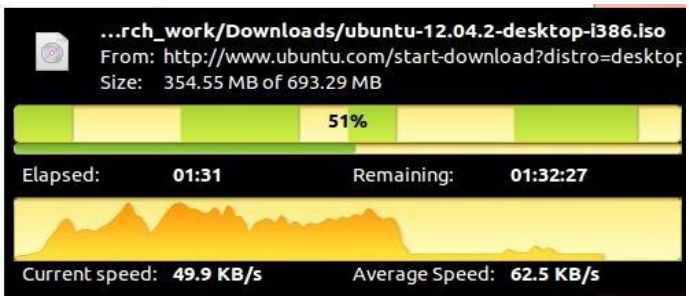


Figure 4.5: Downloading speed after applying bandwidth throttling.

Test carried out before the bandwidth throttling configuration shows that the downloading speed is low(24%) as compared to when bandwidth throttling approach is applied(51%).Moreover, despite the fact that few users were on the network at the time of testing, the speed was still poor. Comparing the network that has applied bandwidth throttling with one that has not shows high level of quality of service. Despite great improvement, a user in the network wanted to access a video tutorial but the site was temporary unavailable which inconvenienced him a lot. From the analyzed results the method achieved 5% redundancy elimination and this was due to the probability that part of the blocked portion contains redundant data.

4.3.2 Packet routing redundancy aware elimination mechanism

We consider network design problems for information networks where routers can replicate data but cannot alter it. This functionality allows the network to eliminate data-redundancy in traffic, thereby saving on routing costs.

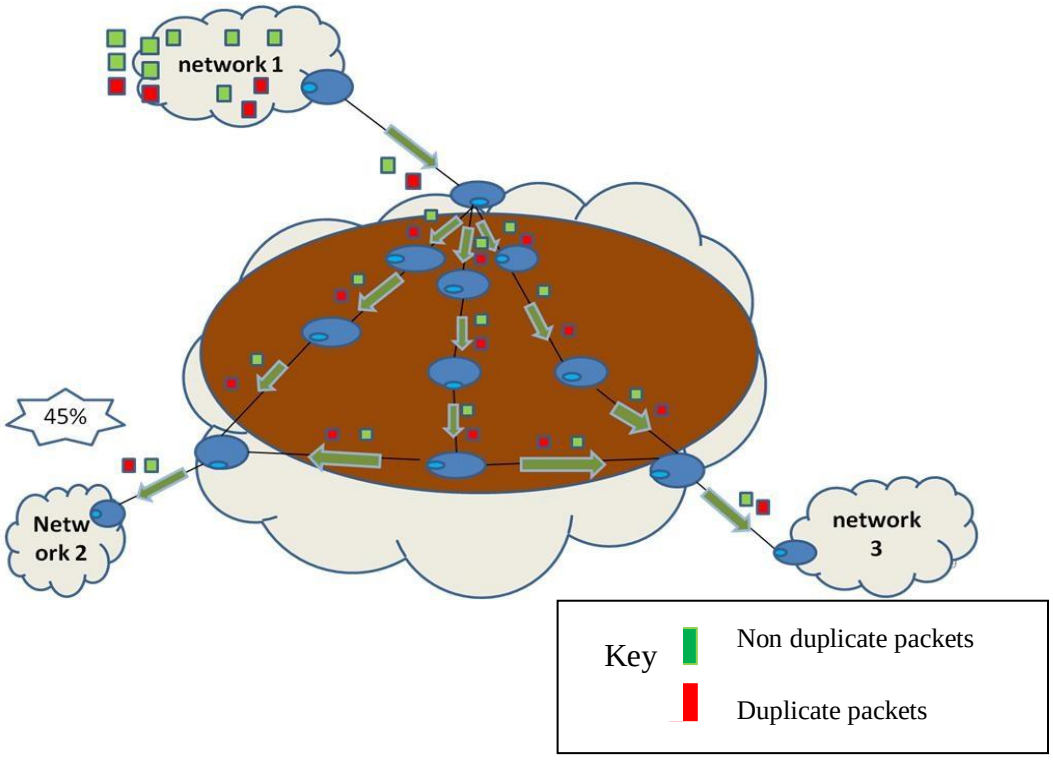


Figure 4.6: Simulation of packet routing

Assuming 12 packets are being sent from router in network 1 to router in network 3 and 2, and then applying redundancy aware routing we will have only transmitted 10 packets instead of 12 at a given route saving 20% bandwidth. Consider network design problems for information networks where edges can replicate data but cannot otherwise alter it. In this setting our goal is to exploit the redundancy in the given traffic matrix to save on routing costs. Formally, we are given a graph over a single server and many clients.

The server has a universe of data packets available, and each client desires a subset of the packets. The goal is to determine a collection of paths, one from the source to each client, such that the total cost of routing is minimized. Here the cost of routing on an edge is proportional to the total size of the distinct packets that the edge carries. For example, if the edge belongs to two paths that each carry the same packet, then the edge only needs to route the packet once and not twice.

However, constructing redundancy aware routes is challenging since it is not be economically viable to deploy redundancy elimination in every link. Also to preserve end to end performance and control signaling costs, routes cannot be determined on a per link basis, routes have to be determined independent of packet content.

4.3.3 Byte level redundancy elimination

If chunking can be applied further to the captured packet, redundancy will be identified as much as possible across the entire content at the source. To accomplish this, we can reuse any of finger printing techniques (Alexander 2010). The chunking process proceeds as follows: for each file, of size L, a window of size w is moved from the beginning of the file to the end of the file. This window w represents the minimum size of the redundant string (or contiguous sequence of bytes) we would like to identify, and it is usually a number between 32 and 64. Each of the total $L - w + 1$ strings are used to compute a Rabin fingerprint generating $L-w+1$ fingerprints.

Using this method bandwidth savings can be represented as function of the minimum chunk size C for the network traces, considering cache size S ranging from 0.25 to 1.5Gbytes.

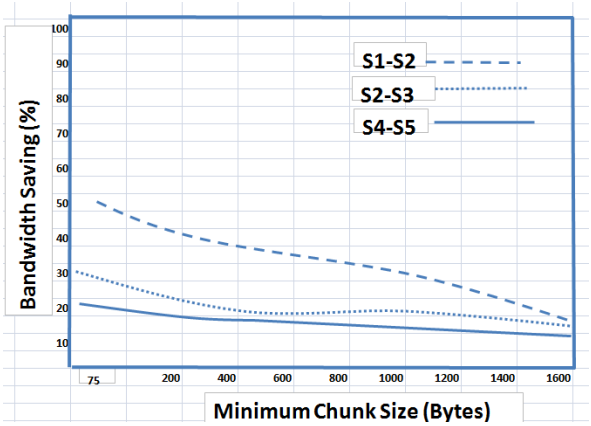


Figure 4.7: Bandwidth saving versus minimum chunk size

Figure 4.7 shows that for both traces the peak of the bandwidth savings results with the minimum chunk size (C=75Bytes).Increasing the value of C has two effects. First, the bandwidth savings are reduced, e.g., only 10% bandwidth saving when C=1425Bytes; this happens because the efficiency of the RE scheme reduces with large chunk sizes, as the probability to find redundant data decreases. Therefore the smaller the chunk size the higher redundancy is eliminated.

However when chunking was done and redundant profiles applied to each file, similarity of the files showed that the binary files were either completely similar to each other, or not similar at all (no visible data points between 0% and 100% similarity). This finding lead us to the observation that applying RE algorithm over binary files does not pay back well, and bypassing all such flows will allow us to achieve virtually same compression rate with much lighter processor load

Therefore if typical binary files such as music, videos and executable binary can be exempted from redundancy elimination operations memory operations will be reduced. This will be taken into account by introducing a variable to identify and exempt non-text files.

4.4 Evaluation of processing load in redundancy Elimination mechanisms

Current redundant elimination mechanism removes unnecessary content from network by stripping off chunks of data from upstream routers which is already in downstream routers. Later reconstruction is done by downstream routers based on headers inserted by upstream routers. In the process every packet has to be chunked which requires massive processing power. Applying our enhanced redundancy elimination mechanism means multimedia files (binary files) will be bypassed from chunking meaning network processing load will be reduced. Using results from table 4.2 which has 11 different files with a total of 84655MB shows that 97% of our sample size contains multimedia files while text files are represented by 3%. Applying minimum chunk size of 75byte derived from figure 4.7 to our content we will have 1,128,733,330 processor cycles using existing redundancy elimination mechanism making an assumption that all the packets were chunked. However applying our enhanced RE mechanism will result into 338619999 processor cycle representing a reduction of 91.74% of network memory accesses as shown in figure 4.8.

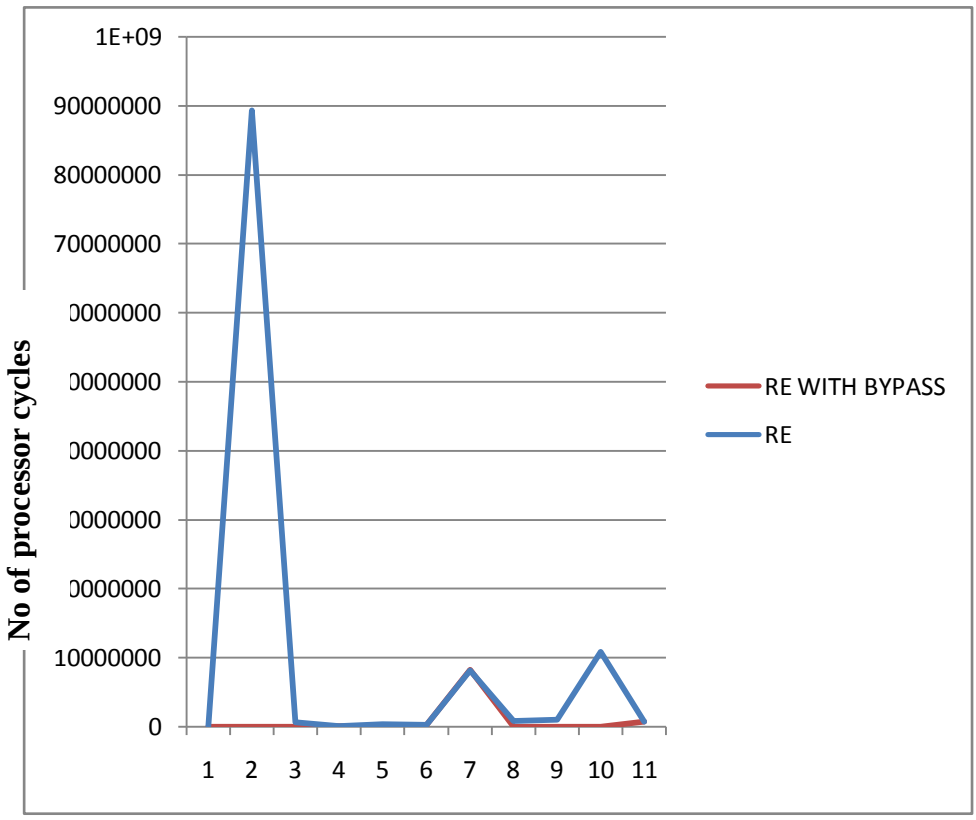


Figure 4.8: Comparison of RE with our enhanced RE.

This shows that our enhanced bandwidth management mechanism is efficient in terms of memory usage (94413333) as compared to existing RE mechanism (1,128,733,330).

4.5 Evaluation of Throughput of bandwidth management techniques

Network throughput refers to the amount of useful data passing through a network. In a network, both useful data and signaling information (that ensures proper reception of the transmitted information) are being transmitted. Of importance to the network administrator is the amount of useful data that his network can accommodate. Networks with greater throughputs are said to be efficient networks. This is because they indicate the proper utilization of the network bandwidth. Therefore if redundant content can be minimized it means there will be high network throughputs.

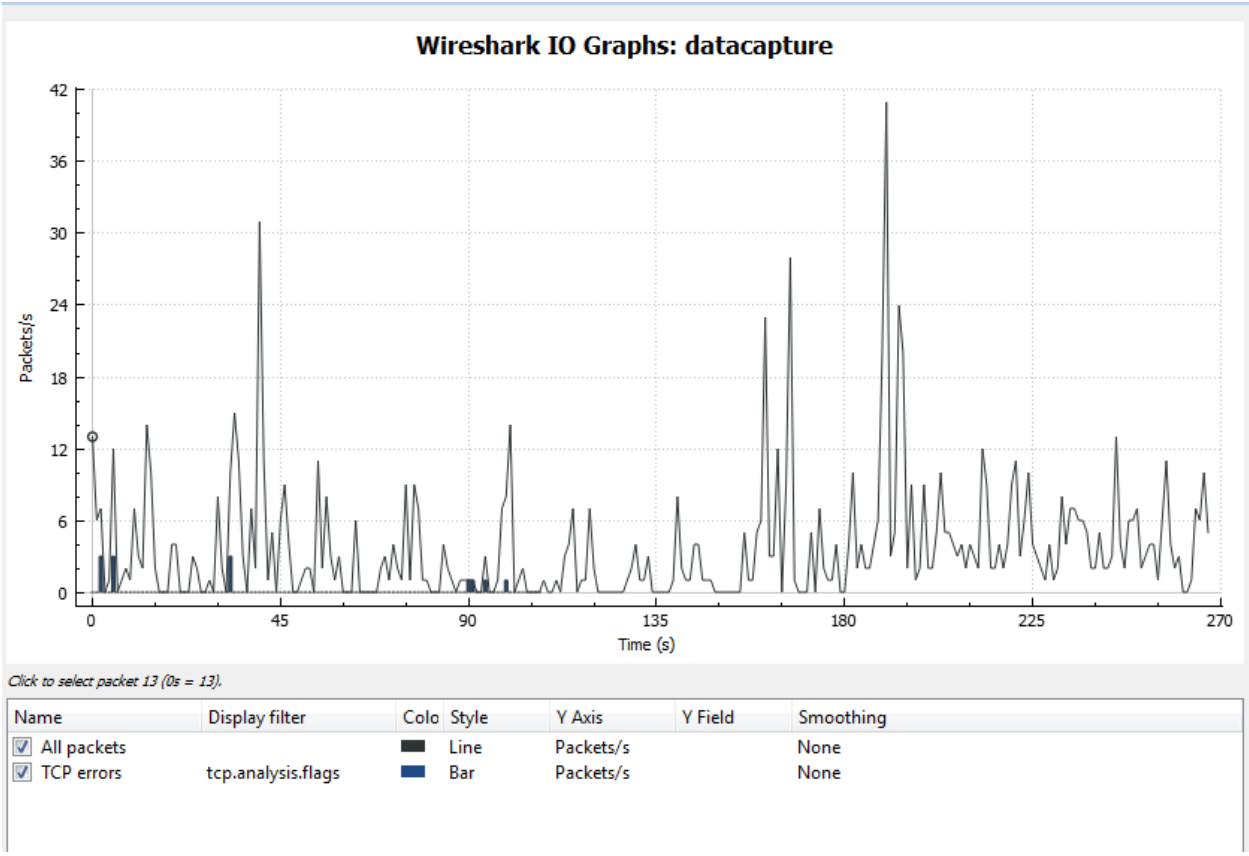


Figure 4.9: Input output data capture graph

the results captured in figure 4.9 shows that network throughput varies over time, with the throughput being higher at one time and very low at other times. For example, 30 S - 45s, 150S-180S there is heavy throughput, while between 90S and 92S, the network throughput is zero. Therefore there is need for coming up with a mechanism that will utilize bandwidth well to ensure maximum throughput.

It is also good to consider the packet window size that will produce best result when deciding network throughput. By applying throughput formula:

$$T = \frac{RWIN}{RTT}, \qquad (4.0)$$

where $RWIN$ is the receive window size and RTT is the Round-Trip Time, it is possible to get the best minimum chunk size that can yield high throughput or maximum transmitted data that can pass through a channel without causing deadlocks.

4.6 Benefits of the Enhanced Information Centric Model

Internet users who purchase bundles from ISPs will benefit by saving costs due to Overall bandwidth saving provided by minimal redundancy mechanism. Initial network costs required to setup an organization network will also be reduced as this mechanism will be implemented as a protocol eliminating purchasing of middle-box devices.

Minimal data redundancy will lead to elimination of traffic jams and denial of service which will make internet applications to be downloading and uploading content at a faster rate. Applications such as Bit torrent, YouTube, Google Video, over the top Video, social networks and photo sharing sites will benefit from this protocol as they require synchronous network setup. These applications will be uploading and downloading more contents at a given time as compared to current information centric networks as traffic jams will be reduced.

Quality of service (QoS) will be high as there will be reduced error rates, better bandwidth saving, increased throughput and minimal transmission delay. These will be important for the transport of traffic with special requirements such as video streaming applications, telephone networks for audio conversations, as well as supporting new applications with even stricter service demands.

CHAPTER FIVE

5.0 Conclusion

In summary, previous researchers have provided valuable and detailed insights into optimization of bandwidth in Information Centric Networks. However, this research recommended deployment of byte level redundant content elimination with bypass to exclude multimedia content from chunking reducing network processing load. This was due the results that showed that our enhanced redundant elimination mechanism would eliminate 46% of redundant traffic a great improvement from the existing mechanisms such as bandwidth throttling (5%) middle boxes (17%), packet routing Redundancy aware routing (21%) and byte level RE without bypass (43%). Our mechanism also proofed to be efficient in terms of memory usage as it reduced 97% of memory operation from existing RE method. Finally this research has presented the promising features of enhanced information centric networks design for the new Internet architecture which are naming, name resolution and data routing, caching, mobility and security. While this work shows benefits of our enhanced information centric model, it is our future endeavor to explore more on the challenges of implementing it as IP layer protocol.

REFERENCES

Alexander A.(2010).*Winnowing: Local Algorithms for Document Fingerprinting.* SIGMOD international conference on Management of data. Chicago: University of Illinois.

Anand A., Sekar, V., & Akella, A.(2009). SmartRE: An architecture for coordinated Network wide redundancy elimination. *ACM SIGCOMM Computer Communication Review. SIGCOMM, 39* (4).

Bengt, A. (2010). Network of Information: information Centric Networking Architecture. German: *Proceedings of the 2008 ACM CoNEXT Conference.*

Bichanga, O., & Wario, Y. (2014). Effects of E-Banking on Growth of Customer Base in Kenyan Banks. *International Journal of Research in Management & Business Studies*. Nairobi: Kenyatta University.

Bonaventure, O.(2011). *Computer Networking Principles Protocols and Practice. Louvain: Université catholique de Louvain.*

Brinkmann, A.(2008). Theoretical Aspects of storage Systems. 12th International Conference, *OPODIS 2008, Luxor, Egypt, December 15-18, 2008. Proceedings.*

Choi, J., Han, J., Cho E., Kwon T.,& Choi Y.(2010). Survey on Content-Oriented Networking for Efficient Content Delivery. *IEEE Comm, 49* (3).

Diego, P.(2012). Redundancy Elimination for Information Centric Networks. *ICN '12 Proceedings of the second edition of the ICN.*

Dina, B., & David, J. (2003). *Duplicate Record Elimination in Large Data Files*. Madison: University of Wisconsin-Madison.

Dolvara, G.(2013) Recent information Centric Networking Approaches. Retrieved from *http://www.cnsm-conf.org/2013/documents/CNSM2013-Keynot2-G-Pavlou.pdf*.

Halgren, B.(2012). Content, Connectivity and Cloud; Ingredients for the future Network. *in Proc. of CoNEXT'09 - 5th International Conference on Emerging Networking Experiments and Technologies, Rome, Italy.*

Justine, S., Shaddi, H., & Arvind, K.(2013).*Making Middleboxes Someone Else's Problem*: Network Processing as a Cloud Service. Washington. University of Washington.

Koponen T., Chawla M.(2007).*A Data-Oriented (and Beyond) Network Architecture*:Kyoto, Japan: John wileys and sons.

Kurose, F., & Ross, W.(2013). *Computer Networks:* A Top Down Approach Featuring the Internet. 6th edition. Brooklyn: Pearson Education printing press

Laura C.(2014). *Wireshark open source software*. Retrieved from https://www.wireshark.org.

Martin, B.(2010). Bandwidth Management and Optimization: Policy development workshop, Nairobi: KENET.

Mironov, I.(2005). *Hash functions: Theory, attacks and applications.* Microsoft Research, *TechReport, Publication no. MSR-TR-2005-187.Chicago,US.*

Mochizuki, .K., Shimazaki, S., Hanawa, D., & Oguchi, K.(2012.) "Proposal of new traffic control method in the next generation home network," in Proc. Telecommunications and Signal Processing, 35th International Conference on Sci. & Technol., Japan: Seikei Univ., Musashino,.

Munir, K. (2014). *An enhanced bandwidth Management scheme for improved quality of service in network communication system*. International Journal of Electronics and Electrical Engineering 01/2014. Retrieved from DOI: 10.12720/ijeee.2.2.147-152.Japan. Cleverland States University.

Nassiuma, K.(2000). *Survey sampling: theory and methods*; Nairobi University press

Neilson, T.(2006). *Photonics for switching and routing*. IEEE Journal of Selected Topics in Quantum Electronics. 19(2).

Oso, W.Y, & Onien D.(2005): *A general guide to writing research proposal and report:* Kampala: Makerere University.

Paulo, M., & Rute, S.(2010). *The role of Information Centric Networking paradigm on the future Internet architectures.* Tokyo: API Interest Group.

Pavlou, G. (2011). *Information Centric Networking overview, current state and key challenges:* London, Pearson printing press.

Rabin, M.(1999). Fingerprinting by Random Polynomials. Technical Report; CRCTTR-15-81, Harvard: Harvard University.

Schmidt, T. (2008). Routing and Packet Forwarding. Finding the Shortest Path. Retrieved from http://www14.in.tum.de/konferenzen/Ferienakademie08/talks/tina_schmidt/presentation_schmidt_tina_routing_and_packet_forwarding.pdf.

Shirley, R.(2014). *The Cryptographic Hash Algorithm Family*: Revision of The Secure Hash Standard And Ongoing Competition For New Hash Algorithms.India: National Institute of Standards and Technology.

Stalling, W.(2007). *Data and Computer Communications*. 8th ed. New York: Pearson/Prentice Hall.

Stolyar, T., & Alexander R. (2010). Shadow-routing based control of flexible multi-server pools in overload. Operations Research, vol3, issue no. 030.

Tanenbaum, S., & Wetherall, D. (2003). *Computer Networks. 5th edition.* Boston Columbus. Prentice Hall:

Tarnauca, B., & Nechifor, S. (2010). *Netinf evaluation*, EC FP7-ICT-4WARD Project, deliverable D-6.3.

Vaidya, R.(2011). Emerging Trends on Functional Utilization of Mobile Banking in Developed Markets in Next 3-4 Years. *International Review of Business Research Papers 7*(1).

Yang, L., & Park Y.(2009) Algorithms and Architectures for parallel processing. 14th International Conference, ICA3PP 2014, Dalian, China.

Zafar, Q., Cheng-Chun, T., & Rui M. (2013). Practical and Incremental Convergence between SDN and Middleboxes. Retrieved from http://nsl.cs.usc.edu/~rmiao/publications/Zafar13_.pdf.