SOFTWARE BASED SCALABLE PACKET CAPTURE MECHANISM TO REDUCE PACKET LOSS

SYAZWINA BINTI ALIAS

UNIVERSITI SAINS MALAYSIA

2019

SOFTWARE BASED SCALABLE PACKET CAPTURE MECHANISM TO REDUCE PACKET LOSS

by

SYAZWINA BINTI ALIAS

Thesis submitted in fulfillment of the requirements
for the degree of
Doctor of Philosophy

June 2019

ACKNOWLEDGEMENT

In the name of Allah, the Most Gracious and the Most Merciful

First and foremost, praise is to ALLAH the Almighty, the greatest of all, on whom ultimately we depend for sustenance and guidance. I would like to thank Almighty Allah for giving me opportunity, determination and strength to do my research.

I would like to take this opportunity to express my sincere gratitude to my supervisor Assoc. Prof. Dr. Selvakumar Manickam for the continuous support of my study and related research, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. Besides my supervisor, I would like to thank NAv6 senior lecturer, Dr. Mohammed F. Anbar for his insightful comments and encouragement, and also for the hard question which incented me to widen my research from various perspectives.

I dedicate this thesis to those who are always in my heart; my late father for his inspirations, my mother for her continuous prayers and endless support, my beloved husband for his great and deep understanding, endless encouragement and sacrifices throughout the study, and my dearest sister and brothers for their continuous support. Without their precious support, this thesis would not have been completed. Thank you very much.

Last but not least, my sincere thanks to Nav6 Director, Prof. Dr. Rosni Abdullah and to all NAv6 centre members and my friends, especially Dr. Layal, Dr. Sabri and Aymen for supporting and encouraging me to be successful in my life.

TABLE OF CONTENTS

CHAPTER 2 - LITERATURE REVIEW

CHAPTER 4 - THE LLSR IMPLEMENTATION

CHAPTER 5 - RESULT ANALYSIS AND DISCUSSION

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

FTP	File Transfer Protocol
IoT	Internet of Things
SDN	Software-defined Network
TCP/IP	Transmission Control Protocol/Internet Protocol
PCAP	Packet Capture
API	Application Programming Interface
UDP	User Datagram Protocol
LLSR	Linked List Storage and Retrieval
DAG	Data Acquisition and Generation
FPGA	Field-programmable Gate Array
BPF	Berkerley Packet Filter
ZCBPF	Zero Copy Berkeley Packet Filter
LIBPCAP	Library Packet Capturing
NIC	Network Interface Card
CPU	Central Processing Unit
ISR	Interrupt Service Routine
DMA	Direct Memory Access
FIFO	First-in, First-Out
NAPI	New Application Programming Interface
MMAP	Memory Mapping
DMA_MAP	Direct Memory Access Memory Mapping
SDK	Software Development Kit

LPF	Linux Packet Filters
OS	Operating System
KPPS	Kilo Packets Per Second
IPv4	Internet Protocol version 4
F	Front
R	Rear
MOD	Modulo
TEMP	Temporary
NMS	Network Monitoring System
RAM	Random Access Memory

MEKANISME PENANGKAPAN PAKET BERSKALA BERASASKAN PERISIAN UNTUK MENGURANGKAN KEHILANGAN PAKET

ABSTRAK

Kemajuan dalam bidang teknologi penangkapan paket telah memberi kesan yang besar dalam proses penangkapan paket di mana kedua-dua penyelesaian perisian dan perkakasan telah direka untuk mengoptimumkan prestasi penangkapan paket. Penangkapan paket telah menjadi lebih mencabar dimana kerumitan penangkapan paket berlaku pada kadar paket tinggi dan jalur lebar tinggi yang mewujudkan keperluan untuk mekanisme penangkapan paket yang lebih sofistikated dalam rangkaian trafik. Tesis ini memberi tumpuan dalam mengurangkan kehilangan paket dalam rangkaian berkelajuan tinggi di mana ia merupakan salah satu masalah yang paling mencabar dalam enjin menangkap paket. Dengan mengurangkan kehilangan paket, ia memberikan kesan yang ketara dalam beberapa aspek dalam prestasi trafik rangkaian terutamanya dengan aplikasi yang berkaitan. Di samping itu, dengan mengurangkan kehilangan paket juga menjadi penunjuk utama untuk menganalisis kualiti rangkaian. Pendekatan sedia ada dalam mekanisme menangkap paket tidak berkebolehan untuk menangkap paket pada tahap berkelajuan tinggi disebabkan oleh beberapa kekangan. Kehilangan paket dalam rangkaian berkelajuan tinggi berlaku disebabkan limpahan penampan di mana kadar ketibaan paket pada penampan meningkat. Tujuan tesis ini adalah untuk mencadangkan dan melaksanakan mekanisme baru dalam menangkap paket untuk mengatasi masalah yang timbul dalam penyelesaian yang sedia ada. Mekasnisme yang dicadangkan adalah untuk mengurangkan limpahan penampan kerana kehilangan paket akan membawa kepada ketidakcekapan prestasi penangkapan paket. Objektifnya adalah untuk merangka

satu mekanisme senarai berpaut untuk meminimumkan kehilangan paket dalam enjin penangkapan paket dalam sistem operasi Linux dan untuk menguji dan menilai stuktur data yang dicadangkan serta membandingkannya dengan pendekatan penagkapan paket yang sedia ada. Pendekatan algoritma yang dicadangkan dikenali sebagai; Linked List Storage & Retrieval (LLSR) menggunakan struktur data senarai berpaut. Pendekatan LLSR dibahagikan kepada tiga (i) penyediaan data (ii) penyimpanan data dan pengambilan semula dan (iii) pemetaan memori. Keberkesanan algoritma yang dicadangkan dinilai dari segi keupayaan stuktur data yang dicadangkan dalam menyimpan paket dengan kehilangan paket yang minimum. Hasil kajian menunjukkan bahawa dengan pendekatan LLSR, peratusan mengurangkan kehilangan paket telah menunjukkan peningkatan sebanyak 32.7 peratus berbanding dengan penyelesaian sedia ada terutama dalam rangkaian kelajuan tinggi. Hasil pendekatan yang diambil dengan jelas menunjukkan bahawa rangka kerja yang dicadangkan dapat mengatasi overflow buffer dan mampu mengurangkan kehilangan paket dalam jaringan berkelajuan tinggi.

SOFTWARE BASED SCALABLE PACKET CAPTURE MECHANISM TO REDUCE PACKET LOSS

ABSTRACT

The advancement of packet capture technology has given a major impact in packet capture process whereby both software and hardware solutions have been designed to optimize packet capture performance. Packet capture has become more challenging by which the complexity of capturing packet occurred in high packet rates and high bandwidth which creates the need for more sophisticated packet capture mechanism in network traffic. This thesis focuses on minimizing the packet loss in high speed network in which it is one of the most challenging problem in packet capturing engine. By reducing the packet loss, it gives a significant impact in several aspects in network traffic performance particularly with related applications. In addition, reducing packet loss also one of the key performance indicators for analyzing the quality of network. Existing approaches in packet capture mechanism is not sufficiently capable to capture packet in high speed network due to some bottlenecks. Packet losses in high speed network takes place due to buffer overflow as their rate of arrival to the buffer increases. The intention of this thesis is to propose and implement a new mechanism in capturing packet to overcome the problem arises in the existing solutions. The proposed mechanism aims to alleviate buffer overflow because packet drop leads to inefficiency of packet capture performances. The objectives are to design a linked list mechanism to minimize the packet loss in the packet capturing engine in Linux operating system and to test and evaluate the proposed data structure and compare it with the existing packet capture approaches. The proposed algorithm approach is known as; Linked List Storage and Retrieval (LLSR)

using linked list data structure. LLSR approach is divided into three steps including (i) data preparation (ii) data storage and retrieval and (iii) memory mapping. The effectiveness of the proposed algorithm is evaluated in terms of the ability of the proposed data structure in storing data with minimum packet loss. The findings demonstrate that by using LLSR approach, the percentage of reducing packet loss has shown an improvement of 32.7% in comparison with the existing solution especially in high speed network. The results clearly revealed that the proposed framework could overcome the buffer overflow and able to reduce packet loss in high speed network.

CHAPTER 1

INTRODUCTION

1.1 Background

Internet is considered to be the world's largest computer network known as a global information infrastructure that connects millions of computers around the world. According to International Communication Union (2015), the number of Internet users worldwide has reached to 3.17 billion which is almost 40 percent of the world population. Internet also has revolutionized the way the world communicates by providing international global, high-speed packet-switched network that offers open and universal access to the general public.

People around the world use Internet as a universal platform for a variety of services including electronic mail (e-mail), browsing the Web, Intranets, File Transfer Protocol (FTP), using social networking applications, accessing multimedia content and services, playing games and many other tasks. Recently, the internet growth is driven by assorted technological paradigm such as Cloud Computing, Internet of Things (IoT), Software-defined Network (SDN) and others. These new technology paradigms envision the next new class of applications and services. Thus, with the emergence growths of the new technology paradigms, the volume of network traffic will increase.

Besides that, with an evolving technology, computer network become more complex and congested due to various factors such as enormous growth in the number of

network users and increasing amount of data transferred. The amount of data moving across a network at a given time refers to network traffic. It is important to have a proper organization of network traffic that helps in ensuring the quality of service in a given network. Network traffic provides benefits in various ways such as identifying network bottlenecks, network security and network engineering.

According to Cisco study (2016), the significant level of traffic growth is driven by five major factors such as: an increasing number of devices, more internet users, faster broadband speeds, more video and Wi-Fi growth. The Internet global growth will result in higher spikes in the network traffic. Therefore, this is the very reason why the technology experts are doing all the best they can to improve the technology especially in overcome the network traffic issues because experts know how vital is internet to people's lives today. Figure 1.1 illustrate the network traffic growth worldwide.

Figure 1.1 Network Traffic Growth

Other than that, network data is mostly enclosed with network packets, which forms the load in the network. Meanwhile, network monitoring traffic is essential to ensure availability and smooth operations. Network traffic monitoring is the process of reviewing, analyzing and managing network traffic for any irregularity or process that can affect network performance, availability or security. In addition, network monitoring incorporates network sniffing and packet capturing techniques in monitoring a network. Network traffic monitoring generally requires reviewing each incoming and outgoing packet.

1.2 Network Monitoring

Network monitoring is very essential in today's modern network infrastructure. Network monitoring is the ability to collect and analyze network traffic (Uma & Padmavathi, 2012). Having network monitoring helps you to visualize what is happening in the network so that any immediate adjustment needed can be done. Network monitoring is useful in observing the usage in the network as well as protecting the network in terms of security. Problems such as low network speed, increase in spam emails and others which can cause frustration in users can be solved by having well - monitored network. The general overview of network monitoring is depicted in Figure 1.2:

Figure 1.2: Network Monitoring Overview

Real time network monitoring enables you to monitor the network in real time and to assess and analyze network efficiency. Typically, network performance is assessed in real time based on the jitter, delay, and packet loss experienced on the network (Cecil, 2006; Uma & Padmavathi, 2012). With real time network monitoring, any issues that happen in the network would alert analyzers immediately and corrective action can be taken to solve the problems. The number of attacks against the Internet-connected system continues to grow at alarming rates. There are two techniques known to quantify exactly what is happening in the network which can be classified as passive technique and active technique.

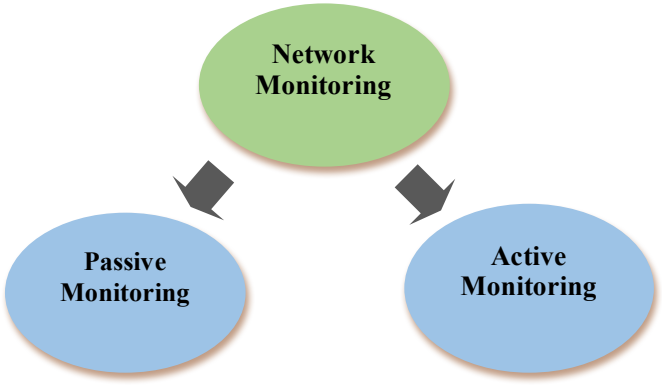

Figure 1.3: Network Monitoring Techniques

Passive monitoring applications analyze network traffic by capturing and examining individual packets passing through the monitored link, which are then analyzed using various techniques, from simple flow-level accounting, to fine-grained operations like deep packet inspection (Papadogiannakis et al. 2012). Meanwhile, active network monitoring transmits probes into the network to collect measurements between at least two endpoints in the network (Cecil, 2006; Uma & Padmavathi, 2012).

1.3 Evolution of Internet Standards

Internet has been evolving from its origin in 1970s until today with a tremendous evolution. In understanding the evolution of Internet, it is important to acknowledge the basic architecture of the system and how does it evolved from time to time. According to Techopedia (2018), Internet standard is a specification that has been approved by the Internet Engineering Task Force (IETF). This specification is important for the interoperability of Internet in terms of consistency and universal use of the Internet worldwide.

Data communications refers to transmission data between two or more computers by means of packets of information travelling over a network. Therefore, in order for these network can work together, they must use a standard protocol or set or rules for transmitting and receiving packets of data. The first set of communications protocols defined in the internet standard is Transmission Control Protocol (TCP) and Internet Protocol (IP).

These internet protocol suite provides end-to-end data communication as well as predates the OSI model, a comprehensive framework for general network system. TCP/IP protocol suites has its own reference model which contains less layer in contrast with OSI model. Meanwhile, the Open Systems Interconnection (OSI) model consists of seven layers that characterizes and standardizes the communication functions of a telecommunication or computing system. Figure 1.4 shows the comparative depiction of OSI and TCP/IP reference model.

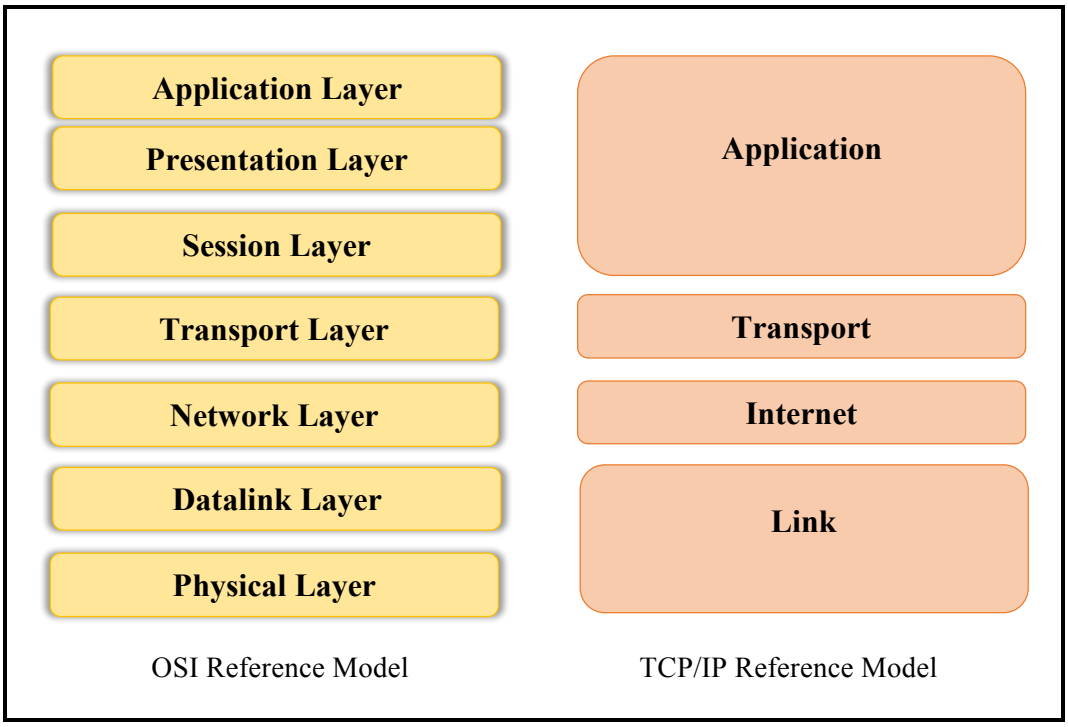

Figure 1.4: The comparative depiction of OSI and TCP/IP reference model.

Internet protocol is one of the major protocols in the TCP/IP protocol suite. Based on the model, it works on the network layer at the OSI model and at the Internet layer of TCP/IP model. Internet protocol version 4 (IPv4) use 32-bit logical address whereby it

provides as a mechanism to uniquely identify hosts by an IP addressing scheme. It also provides the logical connection between network devices by providing identification for each device and route data over the underlying network.

1.4 Need for Packet Capture

Packet capture is defined as you capture the details of the packet that traverse over network. We care about packets for certain reasons such as by having full packet capture, it is useful in resolving the network issues. Besides that, any hidden data can be detected by looking at the raw data that you capture. Different applications will have different reasons of the need for packet capture. Below are the applications and the uses of data capture:

1. **Security**: Network problems and security issues can occur at any time especially when you least expect them. Therefore, it is important to capture the full packet so that a complete assessment can be done and any security flaws and breaches can be determined once you have the complete visibility.
2. **Network Forensics**: When it comes to traffic network monitoring and analysis, you cannot miss any single packets. This is because, in network forensics, by having the complete copy of network activity it allows you to have a quick and clear investigation on what is happening in the network once there is a data breach. In reconstructing the actual network flows, any missing packet in between makes it difficult to be analysed.

3. **Troubleshooting**: Data capture is very important in troubleshooting especially in detecting any occurrence of unwanted events happens in the network.
4. **Higher network performance**: By having 100 percent packet capture of traffic means you can detect a threat or any network performance issues in the real time. Drop packets lead to down network performance.

1.5 Packet Capturing

A packet is a term that was first coined by Donald Davies in 1965, which means a segment of data sent from one computer or device to another over a network. Generally, a packet contains a source, destination, size, type, data and other useful information that helps the packet to get to its destination and to be read. Figure 1.5 shows the diagram of a TCP packet.

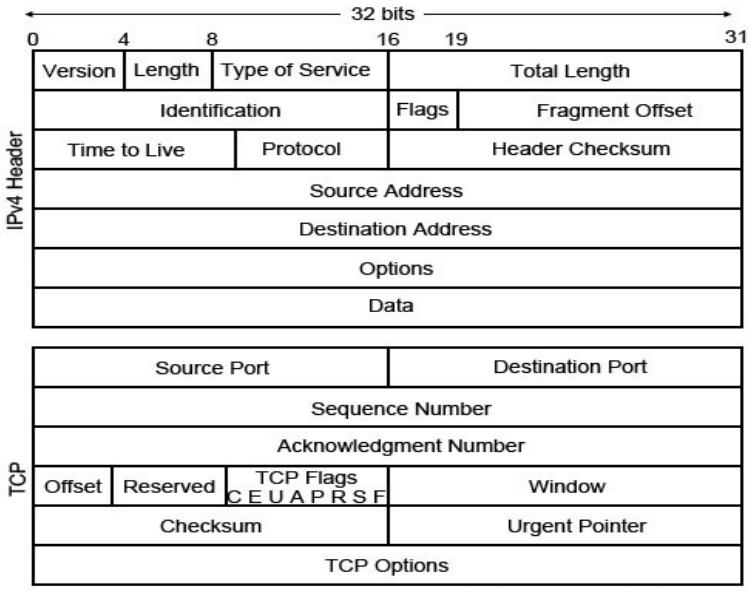

Figure 1.5: TCP/IP Packet

Packet capturing is necessary in any network design and it provides a core capability as a troubleshooting tool. As applications become more complex and networks continue to increase in throughput, it is increasingly important to have the capability of capturing network traffic. In analyzing network capture files, pcap which known as packet capture is one of the security tools consisting of an application programming interface (API). Pcap is implemented as libpcap library in Unix system while in Windows systems, it uses a port of libpcap is known as WinPcap.

Pcap widely used as it is compatible for both incoming and outgoing packets that travel over network. Pcap also has the ability to fully capture complete record of network activity which covers Layer 2 to Layer 7 in the network layer. Therefore, in network traffic monitoring, this library is used to capture packets travelling over network.

Packet loss is acknowledged by most researchers as a critical issue in real-time network traffic. In real-time network especially in packet capturing process, packet loss impacts the overall performance. Whenever there is packet loss in real-time network, retransmission causes delays. Moreover, packet loss can negatively impact the applications whether it uses UDP or TCP packet especially in real-time application by causing quality degradation if the losses are superfluous.

1.6 Research Problem

The major bottleneck in a high-speed network is when the input of packet rate increases. There is a possibility of packet losses happenings as it takes time to process

packet from NIC in kernel space to user space. Circular buffer technology which has been implemented in the existing solutions is no longer suitable to capture packets at high-speed network. Particularly in high-speed networks, packet drop could be lost by the buffer overflow as their rate of arrival to the buffer increases. Therefore, this thesis addresses the following issues:

1. **Packets captured in real-time network are basically in different sizes.**

   This makes it difficult for the packets to be allocated in the memory buffering if the pages are set to fixed sizes. The larger the size of the packets, the fewer the packets that can be queued in the buffer before the buffer fills up. Thus large packets fill the buffer more easily than small packets, and cause the packet drop (Shacham & Mckenney, 2002; Choi, 2012). Different packet sizes also give an effect to the overall performance on certain programming tools-based applications. Research proposed by (Deri, 2004 & Rizzo, 2012) found that memory mapping (mmap) version of libpcap (programming library) is not very efficient in small packets.

a. **By using ring-buffer to allocate new buffers, packets start dropping when they run out of memory or reach certain pre-set limit.**

   In general, circular buffer is an efficient method of temporary storage allocation which entails the rotation data through an array of buffer positions (Budiarto et. al., 2007). The technology implemented through circular buffer enable the data writes advances one step every time new

data is entered into buffer. When the end of the buffer is reached, the process is restarted once again from the beginning of the buffer. However, the traditional way of packet capturing using circular buffer is no longer efficient with the increasing packet loss rate in high speed network. By using ring/circular buffer to allocate or mmap new buffers, packets start dropping when they run out of memory or reach a certain pre-set limit.

2. **Any memory copy degrades overall performance**

   The traditional way of capturing packets needs to be copied twice after being captured. First from network interface to the kernel of operating system and then from kernel to the user space, which leads to the loss of packet capture and reduction in the rate of packet capture. Any memory copy degrades the overall performance. The option to resolve this issue is by allowing multiple threads to work on specific portions of the ring-buffer while it is continuously being filled with new packets and purged by packets processed. In this case, it is quite difficult for analyzers that want to actually keep references to packets for a relatively long time, since it is not feasible without copies out of the ring-buffer into new memory.

1.7 Research Objectives

The objectives highlighted in this thesis are intended to solve the packet capturing engine issues whereby its main objective is to reduce packet loss in packet capturing engine. The specific objectives of this thesis are as follows:

1) To formulate a new data structure to store packet data.
2) To enhance the performance of circular buffer based capture engine by reducing the packet loss.

1.8 Research Motivation

This section discusses the factors that motivate the research undertaken in this research study. First and foremost, the extant packet capture engine is facing packet losses in high speed network. This is discussed in the following sub-section 2.13 and 2.14 that highlighted about the packet losses and the impact of it. Second, the internet traffic has continued to develop at a spectacular rate over the past five years. According to Simon (2017), Internet users have grown by 82 percent, or almost 1.7 billion people, since January 2012. That translates to almost 1 million new users each day, or more than 10 new users every second.

The tremendous of Internet growth have implications towards the network traffic. The volume of internet traffic increases and becomes more intricate. Besides that, even though various solutions have been implemented especially in hardware based which have proven very efficient in capturing packet at higher packet rates, the highly cost in implementing it has leads us to overcome the limitations using software based approach which has lower cost. Thus, this motivate us in improving the technology solutions to overcome the network traffic issues.

1.9 Research Contributions

The current approaches in capturing packets still face the problem of losing packets at a certain rate. The losing packets in real time capturing is crucial in network monitoring because lost packets might contain viruses and worms which will affect the network. The bottleneck in the existing solutions is basically within the capturing chain. It is located along the passing of a captured packet from kernel to the application. The main contribution of this research is a new mechanism that is able to alleviate buffer overflow in the existing solution because packet drop leads to inefficiency of packet capture performances.

1.10 Scope of Research

This research considers in improving packet capture engine in terms of reducing packet loss in order to provide a correct representation of the particular network. The intent of this study is exclusively address the issue in high speed network in which we study the existing technique and proposed the new data structure as the primary solution. The performance of the proposed data structure is evaluated according to dedicated metrics and the outcome is compared to the existing buffering technique.

Besides that, the mechanism was only focused in IPv4 network and mainly at the kernel level in Linux operating system. The discussion is focused on the efficacy and performance of the proposed model in reducing packet loss.

1.11 Research Steps

The research steps that are followed to carry on this thesis are represented as follows:

1. Establish a solid background and literature review of packet capturing engine through discussion, analysis to identify the research gaps of the current buffering techniques used in the packet capture engine.
2. Study the related work and determine the research problem that is going to be addressed in this thesis.
3. Determine the outline of proposed solution that is expected to improve the performance of packet capture.
4. Identify the research methodology of proposed solution which focused on the kernel level. A new module is proposed which is Linked List Storage and Retrieval (LLSR).
5. Derive and analyze algorithm for existing packet capturing process to identify problems that leads to buffer overflow.
6. Evaluate and verified the algorithm for packet capturing process to prove that the assumption about capturing process has caused overflows.
7. Discuss and analyze the obtained results to present the significance of the proposed solution over the existing work.

1.12 Thesis Organization

This thesis is organized into six chapters. Chapter 1 presents the objectives of this thesis, which starts by presenting the importance of Internet as well as a background of

network monitoring. The research objectives, contributions, scope and limitation, and research methodology are also provided in this chapter.

Chapter 2 discusses the most current and related works in packet capture which includes packet capture mechanisms, packet capture architecture and the related existing solutions. The advantages and drawbacks for each approach are highlighted.

Chapter 3 presents the methodology. The proposed solution is introduced in this chapter, as well as the framework of the solution.

Chapter 4 covers the implementation details of Circular Buffer and LLSR for the packet capturing performance

Chapter 5 provides an in-depth analysis and discussion for the LLSR approach.

Chapter 6 provides the conclusion and recommendations for future work.

CHAPTER 2

LITERATURE REVIEW

This chapter presents background information and applicable research related to network monitoring and packet capture in real-time system. Packet capture engine is known as one of the most important factors in network monitoring. Thus, the optimization of packet capture in high speed network is very useful in identifying any bottleneck and potential threat in the network. Section 2.1 provides a brief introduction to packet capture. Section 2.2 provides packet capture solution while Section 2.3 provides a deeper discussion on packet receiving architecture. And in the last section, Section 2.4 presents the existing packet capture software solutions.

2.1 Packet Capture

A packet is the unit of data that is routed between an origin and a destination on the Internet or any other packet-switched network (Issariyakul & Hossein, 2014). Meanwhile packet capture is a mechanism to capture data packets across a computer network. The packet capture process is essential in any network monitoring in order to monitor network traffic, analyze traffic patterns as well as to identify and troubleshoot network problems.

These are among the reasons of having a better capturing engine in order to overcome the issues of packet loss. Other terms for same or similar actions to packet capturing are packet filtering, packet sniffing, network sniffing and packet or network monitoring. In general, there are two types of packet capture which are full packet capture

and partial/filtering packet capture. By using network monitoring application, information in both types of capturing can be easily gathered. Type of packet capture is shown in Figure 2.1.

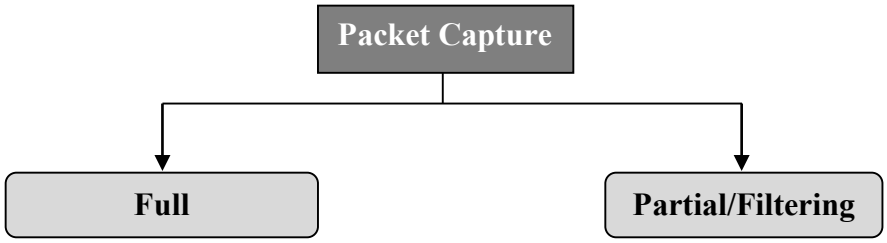

Figure 2.1: Types of Packet Capture

2.1.1 Full Packet Capture

Full packet capture provides the most flexibility and granularity when analyzing network-centric data. Unlike various forms of log data, full content data, if properly collected, is the actual data that was transferred, which is not a summarization, or representation, or sample. A full packet capture can be referred to as the total sum of raw network traffic as it sent from computers and devices on one network to a destination of other network.

Full packet capture is an amazing resource as it can save time whereby you can drill down all the network data from an event. Other than that, capturing full data packet allows an information security analyst to perform detailed network forensics when there is an issue in the network especially in dealing with malicious external threats.

2.1.2 Partial Packet Capture

Partial packet capture contains part of the data being transmitted. It can just record header without recording the total content of datagram. This can reduce storage requirements, and avoid legal problems, yet have enough data to reveal the essential information required for problem diagnosis (Kumar & Arumugam, 2011).

Partial packet capture is useful for smaller capture sizes as it greatly increases the effective storage size. But the disadvantage of partial packet capture is that not all network traffic is stored to disk, and this may bring difficulties in terms of forensics which may be hindered without full data load. Also, data stream reconstruction may not work with partial packet capture.

2.1.3 Packet Loss

Packet losses in high speed network takes place due to various bottlenecks in the packet capturing architecture. In order to address the packet loss problem, it is imperative to locate what actually causes the problems in the packet capturing architecture and identify the casual factors behind them. Software solutions used the same concept of temporary storage allocation whereby most of the packet capturing engines implemented circular buffer technology in their architectures. Architectures that implemented circular buffer in their design are facing packet losses in high speed network due to buffer overflow. Buffer overflow may occur due to a variety of reasons such as (Biswas & Cardigliano et al., 2011):

i. Deficiency of execution between ‘producer’ and ‘consumer’ tasks

ii. Jitter in ‘consumer’ task response

iii. Jitter in the context switching time between the ‘producer’ and ‘consumer’ task

With these overflow problems, using buffer in the kernel as storage is no longer suitable especially in high speed network. Figure 2.2 shows the packet loss comparison for the existing solutions. Beyond Device Polling and nCap can tolerate incoming packet rate up to 570 KPPS. Packet capturing for Beyond Device Polling will be one hundred percent dropped once it reached the maximum packet rate. Meanwhile nCap is able to capture packet better than Beyond Device Polling whereby packets will be lost at the maximum rate of 561KPPS and DashCap captures packets at the maximum rate of 700KPPS without losing any packet.

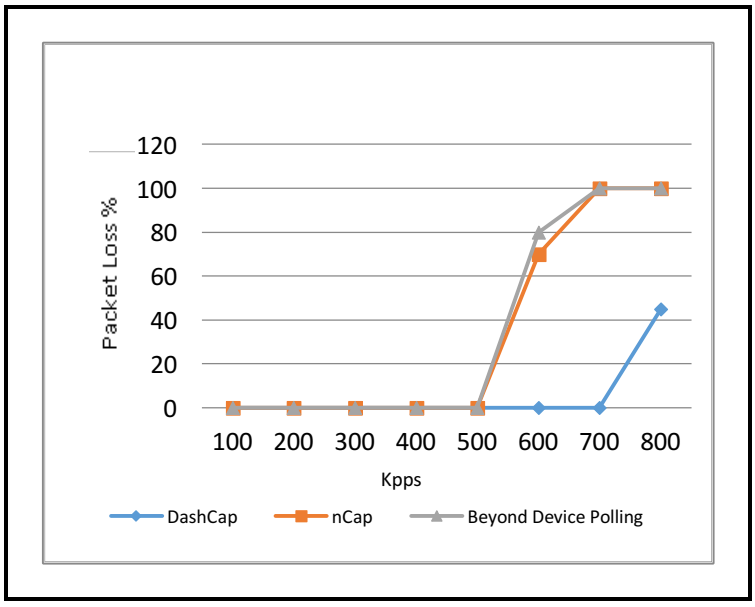

Figure 2.2: Packet Loss Comparison

2.1.4 Impact of packet loss

Reducing packet loss will give major impacts in various aspects. This is because packet loss is an important performance characteristic of network traffic, crucial for applications including long range data transfers, video and audio transmission, as well as video and GRID computing (Friedl et al., 2009). Reducing packet loss is very important because the missing packets could be the key to determining what is happening in the network. Below are the some of the highlights of the packet loss impact:

1. **Video and audio transmission**: When packet loss occurred during the transmission of video or audio, it gives the impact in terms of final service quality.
2. **File transfer and downloading**: Losing packets in the network leads to corrupted file.
3. **VoIP**: Loss packet caused jitter and frequent gaps in received speech.

2.1.5 Factors Behind the Packet Loss

In accentuate that the proposed solution LLSR outperforms PF_RING, it is important to address the issues or factors that impacting the packet loss. Apart from the above mentioned main factors, there are few others that lead to packet loss especially in allocating packets in both solutions. Table 2.1 indicates the factors behind packet loss.

Table 2.1: Factors Behind Packet Loss

Circular Buffer	Linked List
(1) Inserting an element (Packet)	
Packet insertion leads to more number of movements and wastage of memory.	Packet insertion is not depending on the size of the list. It can be done very easily just by the manipulation of addresses.
(2) Deleting an element	
Similar to packet insertion, packet deletion also leads to more number of movements and wastage of memory.	Packet deletion also can be done very easily just by the manipulation of addresses.

2.1.6 Circular Buffer

Most of the existing software solutions use circular buffer in their implementation but it is no longer suitable in today's high speed network. Packet drop could be lost by the buffer overflow as their rate of arrival to the buffer increases. A buffer overflow happens when more data is written to or read from a buffer than what the buffer can hold. This happens in high speed network when the packet rate input increases whereby there is a possibility of packet losses happening as it takes time to process packet from NIC in kernel space to user space.

Buffer is an area of memory used for temporary storage of data when a program or hardware device needs uninterrupted flow of information (Rupasri et al., 2012).

Meanwhile, circular buffer is an efficient method of temporary storage allocation which entails the rotation data through an array of buffer positions. The technology implemented through circular buffer enable the data writes to advance one step every time new data is entered into buffer. When the end of the buffer is reached, the process is restarted once again from the beginning of the buffer. One of the advantages of using circular buffer is that data reading is done in the exact same manner.

Circular buffer contains two index counters to keep track of the current beginning and the end of the queue, also known as Rear (R) and Front (F). The Rear pointer will read information from standard input and place it into the ring buffer. The Front will extract information from the buffer and perform certain operations. Both Rear and Front tasks should be balanced to avoid overflow.

2.1.7 Algorithm for Packet Capturing Engine

The algorithm for packet capturing engine is used as a benchmark to solve the problems that have been identified in existing solutions. Figure 2.3 shows the algorithm for packet capturing engine process. It emphasized the process of inserting packets in the circular buffer slots depending on the condition whether the slots are empty or full. From the process of insertion and deletion of packets, we can see which part the problems are actually occurring and how we utilize it as a benchmark to enhance the packet capturing process. Besides insertion, the process also points out the deletion where the packets are being deleted from the circular buffer and mapped to user space application.

```
While (true) {
        if (packet arrives) check availability of circular buffer;
                if Front = = -1 && Rear = = -1 (Buffer is empty)
                   Return True;
                else
                   Return False;
         else {
                if slot is empty (Front = = Rear)
                   set Front and Rear = 0
                   insert packet in the circular buffer slot as [Rear = (Rear +1) Mod N]
                else
                  if (Rear + 1) Mod N == Front  (Buffer Overflow)
                  Drop the packet;
                else
                   memory mapping;
perform packet and data processing task;
transferred packet to user space as Front = (Front + 1) Mod N ;
        }
}
```

Figure 2.3: Algorithm of Packet Capturing Engine

Based on the algorithm, packets in real-time network arrive at the network card's "First in, First out" (FIFO) receive buffer and the packet is then forwarded from NIC by Direct Memory Access (DMA) to circular buffer. For a given pointer in the circular buffer, Front (F) and Rear (R), the insertion of packet in the buffer is based on the availability of the circular buffer. Figure 2.4 shows the circular buffer diagram with pointer F and R.

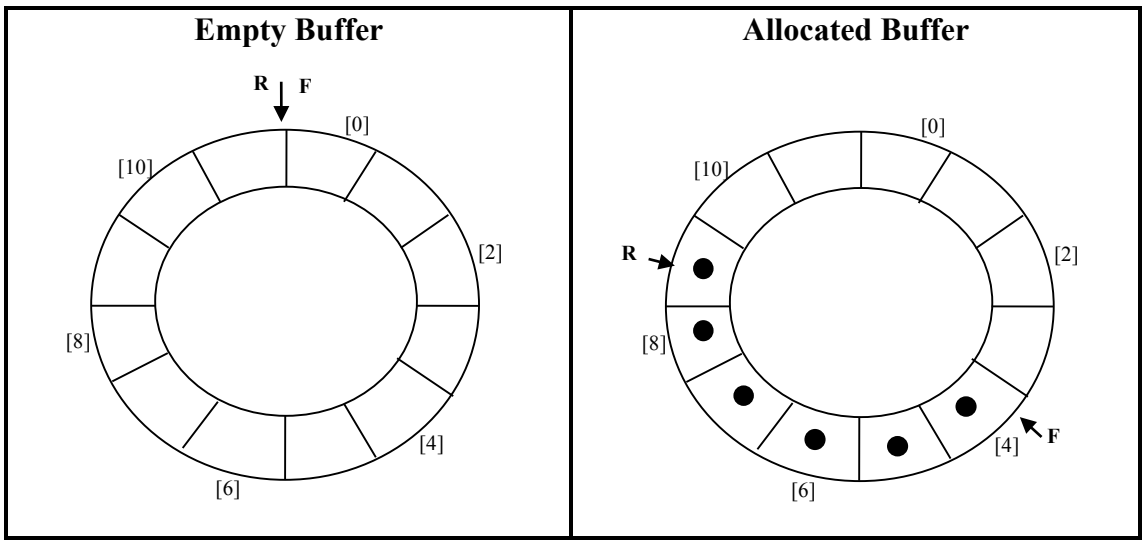

Figure 2.4: Circular Buffer Diagram

As seen in Figure 2.4, by referring to the given pointers F and R, it is assumed that the circular buffer is considered empty where none of the slot is occupied by showing pointer F and R as -1. The insertion of packet in the empty slot is based on the equation of Rear = (Rear + 1) % N where N is the size of the circular buffer.

The problems occurred when we slot in the packets with the equation (Rear + 1) % N = Front which notifies that there is no space for packet allocation. Any packets being inserted will be dropped meaning a buffer overflow has happened. Meanwhile, the packets are deleted from the circular buffer based on the equation of Front = (Front + 1) % N. The stated equations for both insertion and deletion can be considered as an assumption to visualize the real allocation of packets in the buffer slots. From the algorithm, the process of insertion and deletion of packets in circular buffer leads to buffer overflow.