Proceedings of the 6th WSEAS International Conference on Applied Informatics and Communications, Elounda, Greece, August 18-20, 2006 (pp79-86)

ACCURATE AND REAL TIME METHOD FOR NETWORK PACKET CLASSIFICATION

SHAHRIN SAHIB
NORAYU ABD GHANI
FAIZAL ABDOLLAH
Faculty of Infromation Technology and Communication
The National Collage University of Technical Malaysia
Ayer Keroh Melaka
Malaysia

Abstract: - Evolution of Internet has been accompanied by the development of a range of network applications which introduce many interesting, challenging and tricky problem to the network traffic monitoring. The increasing numbers of applications cause network traffic become harder to classify. Classifying internet traffic by looking at headers of packet and the transport protocol port number is no more reliable and accurate technique for the reason in today's computer world, running well-known applications on not-so-well-known ports or used others protocol as wrappers is a widespread scenario. Meanwhile, others examine packet payload in order to identify degree of reliability and accuracy of traffic classification. Yet, due to some restriction such as security, privacy and legal issues, examining the packet payload could be restrained in such cases. However, accurate and real-time methods that can dependably recognize the application that generated a flow still need to be developed without involved in neither human interventions nor offline classification to achieve efficient and effective packet verification. Hence, a common problem with many protocol analyzers is the inability to accurately identify and consequently decode a protocol that runs over a non-default port number. In order to optimize network investment; all packets should be clearly identified; nevertheless, there still no such tool in the market today is capable to analyze all network protocols. In our work, we put on our effort to advance the network monitoring system via an accurate, reliable and robust real time application. In view of the fact that, accuracy in identification and classification of network packet could advance network monitoring, and better understanding of the operational networks applications with the efficient and effective identification of network packet; every abnormal packet could be identified quickly and precisely. In this effort we discovered three main discussion group in packet comparison that different network analyzer will decode packet in different matching set. The depth of protocols defines build packet captured degree of accuracy an issue.

Key-Words: - 1) Packet Classification, 2) Packet Identification, 3) Unknown Network Packet, 4) Network Monitoring, 5) Network Protocol.

1 Introduction

Evolution of Internet has been accompanied by the development of various network applications such as text-based utilities, web, electronic commerce, and multimedia streaming. The main trust behind these activities is the deployment of high-speed long haul communication technologies, which introduce many interesting problem to the design of a network [6] and also monitoring of network traffic facing much more challenge. Of course all these applications causing the growing fraction [6] of all traffic carried by the network and because of that network traffic become harder to classify.

Using specialize network measurement hardware or software, information about network packet could be collected. This information included their timing structure and contents, detail packet-level measurements and some knowledge of the IP stack; they can use reverse engineering to gather significant information about both application structure and user behavior.

All these information can be applied to variety of work like helping in troubleshooting, protocol debugging, workload characterization, and performance evaluation and improvement. We need specific tools to capture and analyze Internet traffic so that data collected from

Internet could be presented in specific pattern for specific purpose wished by an organization. Specific method [7] in measuring Internet traffic need to be clarified to make sure analyzed are meet the objectives.

2 Motivation for this work

Accurate traffic flow classification [2][4] is of elementary importance to numerous other network activities, such as security monitoring and Quality of services, with the intention that the trend of application in operational networks and effectiveness in designing network could be achieved. Yet, classification schemes are difficult to operate correctly because the knowledge frequently available to the network often does not contain adequate information to allow for an accurate methodology.

Remco van de Meent (2004), recent measurements of Internet traffic show a growing fraction of all network traffic is not recognizable. Network traffics is identified by looking at the transport protocol port number however there are some fraction of traffic are unidentified or in other words we still do not know which applications are causing the unknown traffic. In today's computer world, running well-known applications on not-so-well-known ports or used others protocol as wrappers is a common scenario. Thus looking at port number to identify packet is not the best way to reveal the true face of unknown packets. Seeing that, each traffic flow in the network should have an ingress point and an egress point.

Andrew W. Moore et al. examine packet payload in order to identify level of reliability and accuracy of traffic classification. Yet, due to some restriction such as security, privacy and legal issues, examined the packet payload could be restrained in such cases.

Traditional techniques for traffic classification that are often no more accurate than 50 – 70% [4]. In [2] using content-base traffic approached even approaching 100% accuracy however it is extremely resources-intensive process.

As we know some decoders recognize traffic regardless of the port over which it runs, whereas others do not and will define the protocol simply by its lower layer (i.e., TCP or UDP), which in addition means that the decoder does not make available the more valuable field-specific decode information.

Each protocol analyzers have their own capability to analyze every single packet entering the interface with certain numbers of protocols supported. Base on this acknowledge, we could make hypothesis that the number of packet identifiable by each protocol analyzer should be not equal, and the unknown packet classified by each analyzer also varied.

3 Objectives of the Research

Since the changing of the business environment and routine, the network and Internet has become part of our daily lives. New-fangled applications are coming up all the time as people are getting to use the Internet to make their live more cool and easier either in workplace or at home. Ever-increasing in network application will also generate more packets flow in the network. Consecutively to effectively run networks, a good understanding of what is going on the network is required [7].

However, accurate and real-time methods that can reliably discover the generating application of a flow still need to be developed.

The ultimate goal for this research is to accurately identify and classify each packets flow in the network. If any fraction of network packets that unidentifiable existed, it should be dropped or blocked from the network.

We hope with the accuracy in identification and classification of network packet could improve network monitoring, better understanding of the operational networks applications. With the efficient and effective identification of network packet, every abnormal packet could be identified precisely.

4 Related Work

Network traffics classification and identification has maintained most interest for many purposes such as network monitoring, capacity planning, troubleshooting, Quality of Services (QoS) and so on. Normally network packet is classified based on well known port and packet header, this is true in the early days of network development, in today scenario ports number provides limited information. The situations where some of the traffic could not precisely classify that inherit identify network packets are undesirables. Remco ven de Meent, (2004) in his writing recommended that in order to efficiently run networks, a good understanding of what is going on the network is necessary, however the growing of tiny proportion of ‘unknown traffic’ seem possible to clearly identify and classify since more and more new applications using well known port number or well known protocol as binding to make a way into the firewall without being stopped [3].

Unknown traffic also is believed caused by certain type of traffic induces other traffic such as FTP connection induces data transfer that is handled by another connection on different ports. He also suggested the most common way of recognizing applications by purely based on the transport protocol number (listed by IANA and Graffiti) need to be improved. Unfortunately, after applying algorithm developed but without looking at the packet payload, there is still significant fraction of traffic unidentifiable. Although looking at packet payload could cause some sensitive issues and concern of security aspect but we have no choice since often application and users are not cooperative and intentionally or not use inconsistent ports [2].

Andrew W. Moore et al. [3] propose a method which fully use packet payload trace collected from Internet site and attempt to identify the type of errors that may result from port base classification. The author explanation that this technique approaches 100% accuracy however very labour intensive for the reason that multiple classification criteria need to test so as to achieve sufficient confidence in the nature of casual application.

Another work by Andrew W. Moore et al. [3] noted that classification schemes for network traffics are difficult to operate correctly because of insufficient knowledge of the operating network. For example, packet header and port number always does not contain adequate information for a straightforward classification methodology [4]. In this work, a supervise Machine-Learning applied with Naïve Bayes was developed to classify network traffic. However the data tested in this model has been hand classified to one of a number category.

In [5] such effort been done to develop network monitoring architecture to perform traffic capture and processing at full line-rate without packet loss. Performing on-line capture allows application-specific processing and application-specific compression. Combined with a powerful off-line processing interface, this approach can perform full capture to disk of interesting traffic features, yet remains capable of monitoring high-speed networks. However this architecture involved modification of NIC and some enhancement still in development stage to allow processing of other common network applications and thus allow the interaction of such applications with the network and transport protocols to be studied.

5 System Design

Based on IP header behavior of traverse and structure, we develop a system design as follows to accurately identify each network flow so that classification in real-time could be achieved. Datagram at router will be split into smaller pieces as it traverse on the network link that could not handle large frame size, such as WAN, the splitting process called fragmentation which happens at IP layer so everything underneath the IP header will be fragmented.

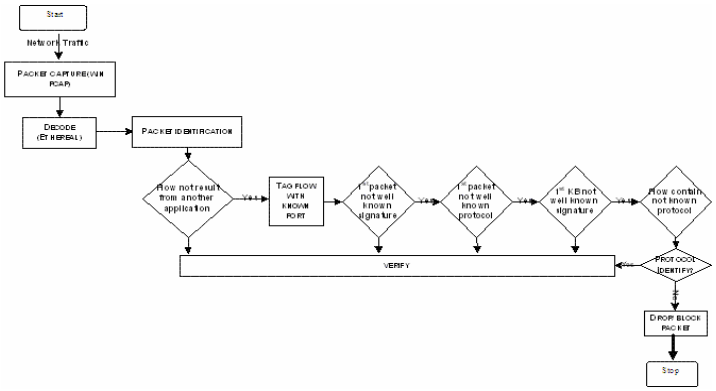
4	8	16	32 bits
Ver.	IHL	Type of Services	Total Length
Identification		Flags	Fragment offset
Time to Live	Protocol		Header Check Sum
Source Address			
Destination Address			
Option + Padding			
Data			

IP Header Structure

Figure 1 IP Header Structure

Each fragmented pieces has it own IP header which is replica of the original datagram header. The Internet Protocol (IP) [RFC791] [8] requires a packet to have three basic elements: source, destination, and data. Thus each fragment has the same identification, protocol, source of IP address and destination IP address as the original IP packet. These elements offer the packet a level of independence. We can straightforwardly identify where it came from, where it's going, and how large it is. All of the fragmented packet are send across the network and will be constructed when arrive at the receiving station.

Andrew W. Moore et al. have develop one mechanism to classify each packet yet still involved human interventions



process make it not possible to be applied for heavy traffic network.

Figure 2 Verification of Flow

We try to solve this classification and identification process without labor intensive involvement. Algorithm bellow describes the processing involved in order to classify each

packet. We obtain basic rules from Andrew W. Moore et al. works with slightly modification by adding block and or drop mechanism at the end of identification process.

```
Pseudo code for classification procedure:
if packet flow not result from another application then
tag flow with known port
if first packet is not well-known signature then{
        if first packet not well-known protocol then{
                if first kilobyte is not well-known signature then{
                        if first kilobyte is not well-known protocol then{
                                if flow not contains known protocol then{
                                        block or drop the packet
                                } verify
                        } verify
                } verify
        } verify
} verify
```

6 Experimental Setup

A Cisco (Catalyst 2900XL/3500XL) switch with Switched Port Analyzer (SPAN) feature is required to make port mirroring available. All the network traffic enter this switch will be captured for analysis by a sniffer using mirroring technique. Packet ingress at source port will be seen by all two sniffers at the same time, so each sniffer will capture traffic under the same condition.

The approach to provide data for the processing system is using on-line data capturing - the data is provided directly to the switch ingress port from the wire. Referring to Figure 3, Catalyst 3500xl switch was used to create one source port in the configuration session and two was set as destination ports, by using the port monitor interface command to create the monitor source port.

Using this system design, we plan to analyze the numbers of unknown packet capture by different analyzer event under same network condition. To meet this target, each sniffer stations are configured to have time synchronous with each others. The number of unknown are further analyze to identify the differences of unknown that captured by each analyzers. The captured packets are applied to the system design architecture to observe the similarity and differences characteristic with each unknown packet.

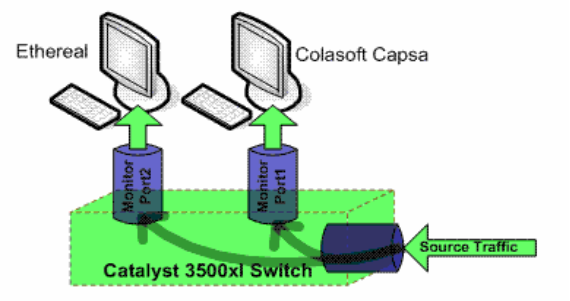


Figure 3 Port Mirroring

In this work we have used two network environments to capture packet traces. First is from measurement on Faculty of Information Technology and Communication in National Collage University of Technical Ayer Keroh Melaka Malaysia (KUTKM) network connecting to the Internet. This network has been chosen because of its different user population and connecting various hardware. Data capture at KUTKM trunk and server farm as second method of application.

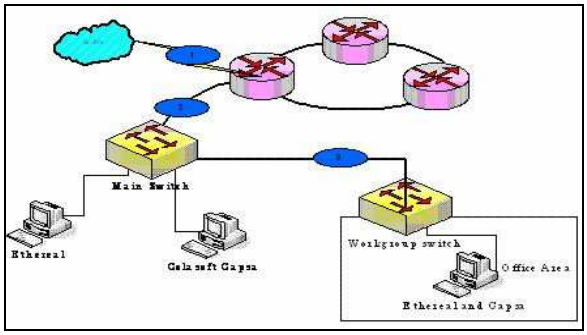


Figure 4 Physical Design

In this setup, we use two sniffer connected to main switch as a comparison of unknown captured by different analyzer.

The source port is configured at main switch interface fa0/1. Traffic from interface fa0/1 is mirrored to another two destination port fa0/2, and fa0/3. One machine with Ethereal and Colasoft Capsa installed is connected to Workgroup switch, traffic from main switch interface fa0/4 is mirrored to fa0/1 at Workgroup switch.

Three points namely Point1, Point2 and Point3 have been identified as the captured tap point. Point1 is traffic from outside KUTKM, Point2 is KUTKM network trunk and Point3 is work area backbone.

To test the reliability of the system design we applied the pseudo code to Ethereal that placed at each point then to compare the numbers of unidentified packet, number of differences.

7 Flow Verification Method

Normally network traffic flows may be classified completely from their initial packet alone [3]. Even so, other flow possibly will need to be examined in more specific and positive identification may be practical up to 1 Kbytes of their data has been observed.

In this work we use attributes listed in [3] and combination with [6] to define and classify each packet:

Attributes	Identification Method
Port base classification source port destination port	Access to the packet header that contain the port number
Packet Header	Access to the entire packet header for both traffic direction
Single packet signature Single Packet Header Signature on the first Kbytes First Kbytes Protocol Flow Protocol (All)	Access to this attribute to examine well-known signature or follow well-known protocol semantic. May require to access to more than single packet's payload
Host history	Already observed-host roles, port scanning, DNS

Table 1 Flow Attribute Identification

We perform simple packet capture at the user station using Ethereal and Colasoft Capsa 5.5 (Capsa) at the same time. From the activity, thirty packets were capture and tabulated Table 1 as bellow.

Table 1 Protocol Decoded by Ethereal and ColaSoft Capsa

Pkt. No	1	2	Pkt. No	1	2	Pkt. No	1	2
1	SSDP	UDP	11	TCP	HTTP Proxy	21	ARP	ARP
2	ARP	ARP	12	TCP	HTTP Proxy	22	ARP	ARP
3	ARP	ARP	13	TCP	HTTP Proxy	23	ARP	ARP
4	IPX SAP	Ethernet	14	HTTP	HTTP Proxy	24	STP	802.2
5	ARP	ARP	15	HTTP	HTTP Proxy	25	SSDP	UDP
6	ARP	ARP	16	STP	802.2	26	BROWSER	NBDGM
7	BROWSER	NBDGM	17	ARP	ARP	27	ARP	ARP
8	ARP	ARP	18	HTTP	HTTP Proxy	28	TCP	HTTP Proxy
9	ARP	ARP	19	TCP	HTTP Proxy	29	TCP	HTTP Proxy
10	ARP	ARP	20	TCP	HTTP	30	TCP	HTTP Proxy

1 – Ethereal Protocol 2 – Colasoft Capsa Protocol

Out of these packets, 36.67% were not identically decoded. Few criteria of discussion could be highlighted as in Table 2.

Given that Ethereal capture info dialog box could not be hold on to after captured, we try to find out which were classified as others protocol by match up to the same packet with other analyzer. Few tables bellow give approaches those comparisons being prepared.

Table 2 Summary of Discussion Highlight

No	Protocol Defined		Discussion Criteria
	Ethereal	Capsa	
1	ARP	ARP	Same definition
2	Browser	NetBIOS Datagram	The depth of protocol definition
3	IPX SAP	Ethernet	Different group of Protocol
4	SSDP	UDP	
5	STP	Ethernet 802.2	
6	HTTP	HTTP Proxy	Different protocol in Ethereal but same in Capsa
7	TCP		

Data presented in Table 1 could be categorized in three main groups as in Table 2 focal discussions, provided that different analyzer have their own suite and matching set of protocol defined; first, the depth of protocol tree defined, second, different group of protocol option, and third different protocol assigned by Ethereal have the same definition in Capsa.

8 Protocol Assignment

8.1 Dept of Protocol Defined

Table 3 (i) Depth of UDP Protocol defined

Packet No 1 - (OTHER)						
SRC ADD	IP	10.1.80.76		ETHERNET	00:C0:9F:9D:62:01	
DST ADD		239.255.255.250			01:0:5e:7f:ff:fa	
CHAR	CHECKSUM	PROT. TREE	PROT.	PRT.SRC	PRT DST	IP.ID
ETHREAL	0x113a	IP	0x0800			0x5dcb
	0xfdf9	UDP	0x11	2365	1900	
		HTTP				
		SSDP				
CAPSA	0x113a	IP	0x0800			0x5dcb
	0xfdf9	UDP	0x11	2365	1900	

Table 4 (ii) Depth of UDP Protocol defined

Table 4 (iii) Depth of UDP Protocol Defined

Packet No 26								
SRC ADD	IP	10.1.80.56		Ethernet	00:04:e2:50:6d:13 (SmcNetwo_50:6d:13)			
DST ADD		10.1.80.255			Broadcast			
CHAR	CHECK SUM	PROT. TREE	PROT.	PRT.SRC	PRT DST	ID	Frame	Signature
ETHREAL	0xe5ab	IP	0x0800			0x9d19		
	0x38d6	UDP NBDGM Service	0x11 Direct_group datagram	138 138	138 Local Master Browser	0x0095	First	
		SMB BROWSER	SMB Mail Protocol Ms Windows Browser Protocol					0 0xAA55
CAPSA	0xe5ab 0x38d6	IP UDP	0x0800 0x11	138	138	0x9d19		
		NBDG				0x0095		
		SMB				SMB		0
		SMB TRANSACTION	Request					

8.2 Different Group of Protocol

Table 5(i) Different Protocol Decoded

Packet No 4 - (OTHER)						
SRC ADD	Ethernet			08:00:37:1b:b8:b3 (Fuji_Xer)		
DST ADD				Broadcast		
CHAR	PROT. TREE	PROT.	PRT.SRC	PRT DST	SAP SERVER Type	SAP SERVER Socket
ETHREAL	IEEE 802.3					
	LLC	0x8137				
	IPX		SAP 0x0452	SAP 0x0452		
	IPXSAP				Xerox 0x076c	0x4f75 unknown
CAPSA	Ethernet SNAP	33079	SAP 0xAA	SAP 0xAA		

Table 5(ii) Different Protocol Decoded

Packet No 16						
SRC ADD	Ethernet			00:06:d7:29:d6:d7		
DST ADD				01:80:c2:00:00:00		
CHAR	PROT. TREE	Type	SAP	Path Cost	ID	Frame
ETHREAL	Ethernet	IEEE 802.3	0			
	LLC	Spanning Tree BPDU	0x42			Unnumbered
	STP	Request			0x8025	
		BPDU - Root		3039	32768	
CAPSA	Ethernet	IEEE 802.2	0x42			

Table 5(i) and (ii) express packet number 4 and 16, two different Ethernet protocol were given. Ethereal packet in frame begins with Ethernet 802.3: LLC: STP (request).

Packet No 25 (OTHER)						
SRC ADD	IP	10.1.80.76		ETHERNET	00:C0:9F:9D:62:01	
DST ADD		239.255.255.250			01:0:5e:7f:ff:fa	
CHAR	CHECK SUM	PROT. TREE	PROT.	PRT.SRC	PRT DST	IP.ID
ETHREAL	0x112f	IP	0x0800			0x5dd6
	0xfdf9	UDP	0x11	2365	1900	
		HTTP				
		SSDP				
CAPSA	0x112f	IP	0x0800			0x5dd6
	0xfdf9	UDP		2365	1900	

However Capsa assigned Ethernet IEEE 802.2.

8.3 Different Protocol definition in Ethereal has same definition in Capsa Analyzer

These following packet declared in Table 6 (i) and 6 (ii), Table , and Table 8 are from the same flow. These packets drawn from packet 11, and 12 where the establishment being agreed and the validation of acknowledge number , SYN flag is turned on to the connection is being established and the sequence number field was chosen by the host (10.1.80.76) for this connection that contain the initial sequence number for the packet.

Table 6(i) HTTP Proxy in Capsa is Defined as HTTP in Ethereal

Packet No 14 – HTTP – HTTP Proxy										
SRC ADD	IP	10.1.80.76		Ethernet	00:0e:84:54:53:3f					
DST ADD		10.1.3.11			00:c0:9f:9d:62:01 (QuantaCo_9d:62:01)					
CHAR	CHECK SUM	PROT. TREE	PROT.	PRT.SRC	PRT DST	ID	Flag	Seq. No	Nx. Seq. No	Ack. No
	0x3123	IP	0x0800			0x5D D0	0x04			
ETHREAL	0x4E5F	TCP HTTP	0x06 Post	8080	2368		ACK PSH	10C8612 6	1157	97738E39
	0x3123	IP	0x0800			0x5D D0	0x18			
CAPSA	0x4E5F	TCP	0x06	8080	2368		ACK PSH	10C8612 6		97738E39
		TCP Option - NOP								

Table 6(ii) HTTP Proxy in Capsa is Defined as HTTP in Ethereal

Packet No 15 – TCP – HTTP Proxy										
SRC ADD	IP	10.1.80.76		Ethernet	00:c0:9f:9d:62:01 (QuantaCo_9d:62:01)					
DST ADD		10.1.3.11			00:0e:84:54:53:3f					
CHAR	CHECK SUM	PROT. TREE	PROT.	PRT.SRC	PRT DST	ID	Flag	Seq. No	Nx. Seq. No	Ack. No
ETHREAL	0x35a8	IP	0x0800			0x9d34	0x04			
	0x8810 0x35a8	TCP IP	0x06 0x0800	8080	2368	0x9d34	ACK 0x04	97738E39		10C865AA
CAPSA	0x63F0	TCP HTTP Proxy	0x06	8080	2368		ACK	977 8E39		10C865AA
		TCP Option - NOP								

Table 7 Ethereal Next Sequence Number

Packet No 18 - HTTP – HTTP Proxy										
SRC ADD	IP	10.1.80.76		Ethernet	00:c0:9f:9d:62:01 (QuantaCo_9d:62:01)					
DST ADD		10.1.80.11			00:0e:84:54:53:3f					
CHAR	CHECK SUM	PROT. TREE	PROT.	PRT.SRC	PRT DST	ID	Flag	Seq. No	Nx. Seq. No	Ack. No
ETHREAL	0x3577	IP	0x0800			0x5dd1				
	0xcb7e	TCP	0x06	2368	8080		ACK,PSH	10C865AA	1204	97738E39
		HTTP								
CAPSA	0x3577	IP	0x0800	2368	8080	0x5dd1	ACK, PSH	10C865AA		97738E39
	0xcb7e	TCP	0x06							
		HTTP Proxy								

Every octet of data sent over a TCP connection has a sequence number. Since every octet is sequenced, each of them can be acknowledged. Ethereal TCP structure in packet 18 contains the Nx Seq. No. (Next Sequence Number) that the sender of the acknowledgement (packet 15) expects to receive which is the sequence number plus 1. This Nx Seq. No. is given for the reason that ACK flag is on and furthermore push flag is on. Push flag is assign for high priority data for the application. For other TCP packet without push flag set the Nx. Seq. No is not appearing.

Table 8 HTTP Proxy in Capsa is Defined as TCP in Ethereal

Packet No 19 - TCP – HTTP Proxy									
SRC ADD	IP	10.1.80.76		Ethernet	00:c0:9f:9d:62:01 (QuantaCo_9d:62:01)				
DST ADD		10.1.80.11			00:0e:84:54:53:3f				
CHAR	CHECK SUM	PROT. TREE	PROT.	PRT.SRC	PRT DST	ID	Flag	Seq. No	Ack. No
ETHREAL	0x3742	IP	0X800			0x9d35	0X04		
	0X63C1	TCP	0X06	8080	2368		ACK	0x97738E39	10C865D9
CAPSA	0x3742	IP	0X800			0x9d35	0X04		
	0X63C1	TCP	0X06	8080	2368		ACK	0x97738E39	10C865D9
		HTTP Proxy							

9 Discussion

Fragmentation of packet at the router could cause increase the probability of that packet will not arrive in order at the receiving station. Some time, some of the fragments were lost while sending them from the source to the destination. Packet also could be damage due to several factors such as:

- hardware malfunction
- broken link
- discard by receiving buffer and
- packets lost in burst

Damaged done to packets will reduce the probability to accurately identified each packet. Since every analyzer have their suite of protocols supported and vary from one to another, so we could expect that the classifications of network packets would not be identical with the use of different protocol analyzer. Consequently this resulted different protocol is assigned to the same packet.

Refer to Table 1, ARP protocol is well defined by both Ethereal and Capsa. For packet no.1 and no.25, two distinct protocols group were allocated. SSDP – Simple Service Discovery Protocol is used by Windows messenger to attempt to locate upstream Internet gateways on Port 1900 as in packets no.18 and 19, Ethereal defined SSDP as sub protocol in UDP with protocols in Frame structure's begin with - Ethernet: IP: UDP: HTTP whereas Capsa assigned UDP. Capsa presented HTTP Proxy protocol for packet no.11 – no.15, no 18 and no.19 as well packet no.28 to packet no.30, as a caparison, Ethereal group packet no.11, 13, no 19 and no.28 to no.30 as no more than TCP, while packet no.14,15 and 18 as HTTP. HTTP proxy protocol has been captured seeing that proxy server being applied to the directed network.

Within Ethereal, IPX SAP is defined as Service Advertisement Protocol which is in frame of Ethernet: logical link control (LLC): Internet Exchange Protocol than Service Advertisement Protocol (SAP). Ethernet protocols refer to the family of Local Area Network covered by IEEE 802.3; it is mainly used in Novell NetWare 2.x and 3.x networks. Look into packet number 4, Ethereal and Capsa have different view for the same packet from the beginning, Logic Link Control (LLC) is the IEEE 802.2 LAN protocol that specifies an implementation of the LLC sublayer of the data link layer. IEEE 802.2 frames [10] [11] contain fields similar to the Ethernet 802.3 frames with the addition of three Logical Link Control (LLC) fields. Novell NetWare 4.x networks use them and IEEE 802.2 is used in IEEE802.3.

Packet number 4 have different protocol assigned and put under Ethernet SNAP (Sub-Network Access Protocol other protocol) by Capsa, Ethernet SNAP frame type builds on the 802.2 frame type by adding a type field indicating what network protocol is being used to send data. This frame type is mainly used in AppleTalk networks. Ethereal assigned this packet as Ethernet 802.3.

10 Finding

As shown in section 8.1, two analyzer show different criteria in decoding network traffic packet as for the same packet, different protocol name is assigned, and even some packet classified by one analyzer but not the other analyzer. If the analyzer could not match the criteria carry by the packet, then the packet was grouped under unknown protocol (Packet 1, 4 and 25 - Other protocol by Capsa). Different capability to decode packet may have the probabilities of one packet is code as unknown. Packet 1 and 25 were from UDP packet while packet number 4 was Ethernet.

From this simple capture activity, three characteristic were discovery, first, the dept of protocol tree defined, second, different group of protocol option, and third different protocol assigned by Ethereal have the same definition in Capsa. This distinctive protocol definition may caused by different protocol definition in capture library.

11 Conclusion

Increasing numbers of applications cause network traffic harder to classify. In view of the fact that the sole use of packet header and port number no more reliable as packet identification and classification method. Although examine packet payload could achieve nearly 100 percent accuracy in traffic classification, however we need one such method that no require human intervention and could be done real time to classify and identify network packets. On the other hand, due to some restriction such as security, privacy and legal issues, examined the packet payload could be restrained in such case.

Each packet running on the network need to be identified in order to efficiently use network resources such as bandwidth, and

effectively network monitoring activities, furthermore every abnormal packet could be identify quickly and accurately. As claimed by some researcher it is possible to verify every packet in the network, thus base on this work new technique will be purpose as a based line to develop an accurate, reliable and robust real time application which could be enhance the network monitoring system.

For every packet flow, protocol decoder attempts to find a match in its defined set of protocols and subprotocols. If a match cannot be found, then the packet is marked as "unknown". The subprotocol is the next layer down within each protocol. In this work, the unknown subprotocols for each protocol are to be further group so that it provides meaningful classification of each packet flow.

Packet classification is an enabling technology for next generation network services and often the primary bottleneck in high-performance routers as well as for monitoring purposes. Packet classification is important for applications such as firewalls, intrusion detection, and differentiated services [9].

12 Future Work

To realize the full potential of the traffic classification more work still need to be done especially on identifying related feature inside network traffic and how they are interrelated. The data generated from the network traffic monitoring tend to have very high volume, dimensionality and heterogeneity making the performance of the traffic classification unacceptable for the on-line analysis. Furthermore, there are many features inside network traffic but which feature is useful for the classification, why they are selected and how there are correlated to each other still need to be discovered. Therefore the ability to identify the important feature and eliminate the redundant feature will help to increase accuracy and achieve maximum performance in classifying the traffic especially on detecting the intrusion inside the network. Therefore in the next research we will concentrate on answering the above question especially in implementing a real time application.

References:

[1] Ramon Caceres. (1989). Measurements of Wide Area Internet Traffic. Technical Report, CSD-89-550, University of California at Berkeley.

[2] Thomas Karagiannis, Konstantina Papagiannaki, Michalis Faloutsos. (2005). BLINC: Multilevel Traffic Claffication in the dark. In Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications SIGCOMM '05.

[3] Andrew W. Moore and Konstantina Papagiannaki. (April, 2005). Toward the Accurate Identification of Network Applications. Intel Research Laboratory, Cambridge. In Passive & Active Measurement Workshop, Boston, U.S.A.

[4] Andrew W. Moore and Denis Zuev. (June 2005). Internet Traffic Classification Using Bayesian Analysis Techniques. In proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems SIGMETRICS '05, Volume 33 Issue 1.

[5] A.W. Moore, J. Hall, C. Kreibich, E. Harris, and I. Pratt. (April 2003). Architecture of a Network Monitor. In Passive & Active Measurement Workshop 2003 (PAM 2003), La Jolla, CA.

[6] Remco van de Meent and Aiko Pras. (2004) Assessing Unknown Network Traffic. University of Twente, Enschede, The Netherlands.

[7] Carey Williamson. (Nov. – Dec. 2001). Internet Traffic Measurement. Internet Computing, IEEE, volume 5, issues: 6, pages: 70 –74.

[8] Internet Protocol DARPA Internet Program Protocol Specification (September 1981), URL http://www.ietf.org/rfc/rfc0791.txt

[9] Florin Baboescu and George Varghese, Member, IEEE, (February 2005), Scalable Packet Classification, Ieee/Acm Transactions on Networking, vol. 13, no. 1.

[10] RFC 1042

[11] RFC 894