Flow Based RED for LAN Useful Traffic Classification and Detection To ensure Fair Bandwidth Usage

Yihdego, Abel Nahusenay
Dept. of Telecommunications Engineering
University of Trento
Trento, Italy
abel.yihdego@studenti.unitn.it
abelnahusenay@gmail.com

Abstract— Traffic classification is an automated process which categorize computer network traffic using various parameters like port number or protocol in to a number of traffic classes. Each resulting traffic class can be treated differently in order to differentiate the services implied for the user. In this literature I am suggesting to make Random Early Discard flow based and use it to study traffic entering a local area network by performing Passive Measurement and Identifying the useful responsive and unresponsive flows so that not to penalize all unresponsive flows.

Keywords— *Random Early Discard (RED), Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Internet Protocol (IP), Source (SRC), Destination (DST), Internet Control Message Protocol (ICMP), First In First Out (FIFO), Minimum Threshold (min-th), Maximum Threshold (max-th), Deep Packet Inspection (DPI), Quality of Service (QoS), Passive Measurement (PM) Congestion Avoidance (CA), Congestion Control (CC), Scheduling Algorithm (SA), Voice Over IP (VOIP), For Example (E.g).*

I. INTRODUCTION

Network traffic classification is a technique used to classify network traffic based on features passively observed in the network according to specific classification goal. It has a wide range of applications in network security and management, such as Quality of service control, lawful interception and intrusion detection. However, classification schemes are difficult to operate correctly; because packet headers often does not contain sufficient information to allow for an accurate methodology.

Network traffic classification is achieved by various means. The easy approach is to use Port Numbers, which is a fast, low resource consuming and supported by many network devices. But it is only useful for applications and services, which use fixed port numbers and it is easy to cheat by changing the port number in the system. Another method called DPI is a signature based classification approach, Inspects actual payload of packets, detect applications and services when they operate regardless of port numbers. But DPI requires a lot of processing power, has poor performance for Encrypted applications and signatures must kept up to date since applications change very frequently. The other approach called Statistical classification, relies on good statistical analysis (PM) of attributes such as packet size, packet inter-arrival times and byte frequencies. This technique is fast compared to port based classification and can detect class of yet unknown applications compared to DPI.

Therefore, to classify network traffic in a given LAN and propose a solution to aggressive flows and solve issues of abnormal flows, someone has to let in every traffic first and has to perform analysis without causing congestion in the network. Here I am using the One byte IP layer protocol field (TCP=06, UDP=17 and ICMP=01) to classify flows at router level (the point at which traffic enters the LAN network) using flow based RED [3B]. And later perform analysis using the remaining 4 tuple information's (SRC and DST ports, SRC and DST IPs) together with statistical analysis stated above.

II. LITRATURE REVIEW

Allowing every incoming traffic to enter a given network and performing PM to identify useful UDP and TCP applications, without causing congestion in the network is a critical task.

The IP depends on the CA mechanisms implemented in the transport layer protocols, like TCP, to provide a connectionless, best effort, end-to-end packet delivery service under heavy load. However, a lot of TCP implementations do not include the CA mechanism either deliberately or by accident. Moreover, there are a growing number of UDP based applications running on the internet, such as voice and video packets. These flows of these applications do not back off properly when they receive congestion indications. Therefore, in normal cases it is necessary to have router mechanisms to shield responsive flows from aggressive flows.

There are two types of router algorithms for achieving CC, the generic SA and queue management algorithms. SAs are exemplified by Fair Queuing algorithm, which requires buffer at the output of a router to be partitioned in to separate queues each of which will buffer the

ISSN: 2229-6913 (Print), ISSN: 2320-0332 (Online)

packets of one of the flows. Here packets belonging to different flows are essentially isolated from each other and one flow cannot degrade the quality of the other. However, this approach requires a complicated per flow state information, making it too expensive to be widely deployed.

Recently, to reduce the cost of maintaining flow state information, SA called Core Stateless Fair Queuing (CSFQ) is used. Here router are divided into two categories: edge routers and core routers. An **edge router** keeps per flow state information and estimates each flow's arrival rate. These estimates are inserted into the packet header and passed on to the core routers. A **core router** simply maintains a stateless FIFO queue and during period of congestion, drops a packet randomly based on the rate estimates. This scheme reduce the core router design complexity, however edge router`s design is still complicated. Thus, scheduling algorithms can provide a fair bandwidth allocation, but they are often too complex for high-speed implementations and do not scale well to a large number of users. On the other hand, queue management algorithms have had a simple design, and the hope is to approximate fairness. This class of algorithms is exemplified by RED. A router implementing RED maintains a single FIFO to be shared by all flows, and drops an arriving packet at random during periods of congestion. The drop probability increases with the level of congestion. Since RED acts in anticipation of congestion, it does not suffer from the lock-out and full-queue problems of earlier Drop Tail. By keeping the average queue size small, RED reduces the delay experienced by most flows. However, like Drop Tail, RED is unable to penalize unresponsive flows. This is because the percentage of packets dropped from each flows over a period of time is almost the same. Consequently, misbehaving traffic can take up a large percentage of the link bandwidth and starve out TCP friendly flows.

To improve RED's ability for distinguishing unresponsive flows, few variants was proposed (like RED with penalty box) and Flow Random Early Drop (FRED). However, these variants incur extra implementation overhead since they need to collect certain types of state information. RED with penalty box stores information about unfriendly flows while FRED needs information about active connections. An algorithm called Stabilized RED (SRED), stabilizes the occupancy of the FIFO buffer, independently of the number of active flows and finds candidates for misbehaving flows. Although SRED identifies misbehaving flows, it does not propose a simple router mechanism for penalizing misbehaving flows. Later proposed algorithm called CHOKe simultaneously identifies and penalizes misbehaving flows by dropping more of their packets. By doing this, CHOKe (CHOose and keep for responsive flows, CHOose and Kill for unresponsive flows) aims to approximate maximum and minimum fairness for flows that pass through a congested router.

III. MOTIVATIONAL PROPOSAL

A. Problem Formulation

The goal here is not to penalize all unresponsive flows (UDP flows) in a given LAN network. Because some of them are very important to making life so easy. E.g. VOIP applications like (Viber, Vonage, Tango, Line, WeChat and Whatsapp), Web Radio or TV (audio and video streaming), video conferencing and so on.

B. Methodology

To perform traffic classification, studding every incoming traffic to your network by performing analysis (PM) is mandatory. To do so I am using the advantage of a Queuing Management Algorithm: RED being unable penalize unresponsive flows. This is important to identify **useful** UDP and TCP applications, from a vast traffic without causing congestion in the network. A router implementing RED maintains a single FIFO to be shared by all flows, and drops an arriving packet at random during periods of congestion.

Flow Based RED

Routers implementing RED keep track of the average queue size, and mark two thresholds: the **min-th** and **max-th.** Look figure-1 below.

1. If the average queue size is below **min-th**, a new arriving packet is checked it's IP protocol field immediately (TCP **06**, UDP **17** or ICMP **01**) and queued in either of the three FIFOs, and forwarded to switch.
2. If the average queue size exceeds **max-th,** the incoming packet is dropped.
3. If the average queue size is between the **min-th** and **max-th**, an arriving packet checked it's IP protocol field (TCP **06**, UDP **17** or ICMP **01**) and is queued in either of the three FIFOs and at last forwarded to a layer a switch (layer three switch).

Here I am **modifying RED** to be a flow based, at least to have three FIFOs for the widely used transport layer protocols (TCP, UDP and ICMP). This step is vital to study the various applications, services and protocols transported by these transport layer protocols. To perform a separate study of these flows, I added a **filtering algorithm** to the normal RED algorithm that checks the

ISSN: 2229-6913 (Print), ISSN: 2320-0332 (Online)

one byte IP protocol field located in the ninth (9^{th}) byte of the IP header (TCP=06, UDP=17 and ICMP=01). This will have little overhead over RED but is essential to classify flows and perform analysis independently to packets coming using TCP and UDP.

C. *Passive Measurement and Principles of Protocol Detection*

To perform PM, the output of each 3 flows in a figure-1 below, goes to a layer two or three switch and make the analysis of incoming packets from an end device (computer or server) installed with Wire-shark connected to the switch. **Wireshark** is one of the best open source packet analyzers available today; packets can be captured live from a network interface and saved for later analysis.

Performing a pre analysis using features of Wire-shark, by generating own traffic of the useful VOIP applications mentioned above [A]; a signature can be extracted that perfectly characterize the protocols or applications. And later by performing analysis on the real incoming traffic, the more aggressive UDP flows **other than the useful ones** can be identified with a proper signature. This is a vital step to ensure fair bandwidth usage to other TCP flow in a given network. Using the signatures (sets of rules derived from packet analysis result), the Firewall near to the LAN can be adjusted to control (block) incoming traffic. This way Fair Bandwidth Usage can be ensured in a given LAN.

Protocol Detection

Good Protocol Detection requires perfect analysis (PM) to extract a meaningful signature from packet information; it can be signature based or behavioral. Signature based approach focus on packet header information's such as:

- SRC or DST Port and IPs.
- Transport layer Protocol (TCP or UDP).
- Specific Patterns (E.g GET, HOST in HTTP Protocol).

The Behavioral Approach, looks for attributes such as (average packet size, packet inter-arrival times and byte frequencies).

D. *Analysis Result*

I captured packets of live sessions for Viber (one among The VOIP Protocols). And here are the analysis results of the offline packets.

VIBER: use's TCP ports 4244 and 5242 for Instant Messaging (IM) and use's UDP port 7985 for VOIP most of the time. On the data part of each packet (next to the TCP header), the following hexadecimal byte values are always the same (P[1] =0X00, P[3]=0X00, P[4]=0X00). And on the data sections next to UDP, the packets have always either of following fixed sizes from the start of the session to termination (12, 20, 21, 44, 104, and 126) bytes.

In the same manner, certain rules can be extracted after analysis to control the flow of aggressive traffic and other abnormal flows in a given network.

E. *Algorithm flow diagram*

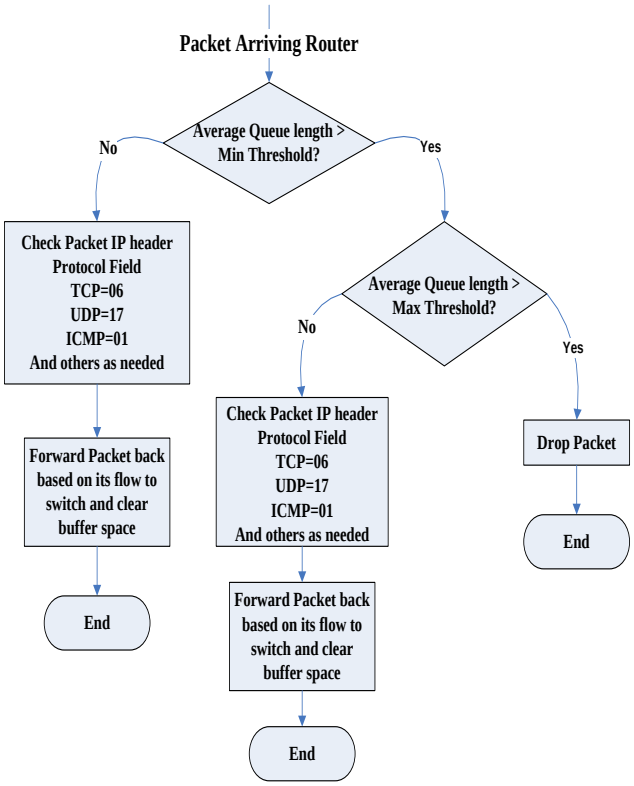

Figure-1. Flow Based RED

IV. CONCLUSION

To sum up, by adjusting the **Firewall** leading to the target LAN, based on the sets of rules derived from packet analysis result (signatures like Viber's case above); it is possible to control (block if necessary) aggressive and abnormal flows from the incoming internet traffic. This way Fair Bandwidth Usage can be ensured in a given LAN.

REFERENCES

[1] http://www.firewall.cx/networking-topics/protocols.html

[2] Rong Pan, Balaji Prabhakar, Konstantinos Psounis "CHOKe a stateless active queue management scheme for bandwidth

ISSN: 2229-6913 (Print), ISSN: 2320-0332 (Online)

allocation" Department of Electrical Engineering, Stanford University, CA 94305 June 12th 1999.

[3] M.Tamilkili "A survey on recent traffic classification techniques using Machine Learning Methods" Dept. of CSE and Karunya University, India.

[4] Andrew W.Moore and Denis Zuev "Internet Traffic Classification using Bayesian Analysis Techniques" university of cambrige and university of Oxford respectively.

[5] Sally Floyd and Van Jacobson , "Random early detection Gateways for Congestion Avoidance" Lawrence Berkeley Laboratory, University of California.

[6] http://www.firewall.cx/networking-topics/firewalls.html

ISSN: 2229-6913 (Print), ISSN: 2320-0332 (Online)