Reduction of Table Flow Occupancy and Packet Loss Detection in SDN Switch

Mr. VijayGokul S 1, Mr. Dani Regan Vivek J2, Naresh Kumaar B 3, Vignesh N 4

1 Assistant Professor, Department of Electronics and Communication Engineering, Mepco Schlenk Engineering College, Sivakasi – 626005. Sivakasi, India
2 Student, Department of Electronics and Communication Engineering, Mepco Schlenk Engineering College, Sivakasi – 626005. Sivakasi, India.
nareshbala024@gmail.com, haseevignesh4149@gmail.com

Abstract— Software Defined Networking (SDN) facilitating network management and monitoring experiences some inevitable problems which affects the performance in terms of delay, on revisiting the limited flow table size problem, a major roadblock to individual flow management of all flows in SDN. We explicitly defined the objective and cost functions that we are interested in, and presented the optimum (i.e., best theoretically possible) solution for flow promotion to achieve it. Then we used the optimum solution as an inspiration to propose a new flow promotion that predicts TCP flow termination (expedited eviction) and incubates non- TCP flows (delayed installation). We provided simulation results that compared our new proposed delayed installation and expedited eviction approach with the default Open Flow approach, the optimum approach, and size-based and rate-based heavy hitter detection using fuzzy model. The results demonstrate that not only our approach provides significant savings in comparison to the default Open Flow approach, it can also perform better than heavy hitter detectors when we use small idle timeouts.

Keywords— Flow Table Occupancy, SDN, Fuzzy-Logic, Heavy Hitter Detection, Open Flow approach

1. INTRODUCTION

SDN (Software Defined Networks) has grasped the attention of people due to its flexibility to take up the programmability in current networking architectures. It allows quick and simpler network modernism. SDN is used for separating the data plane and the control plane, and makes software implementation of advanced networking applications easy with an advantage. Older network architectures are not useful in establishing the modern concepts. Thus, SDN is made advance by introducing a new platform ONF (Open Networking Foundation). In SDN architecture all the planes (Data and Control) are centralized by a controller. The Data and Control planes are decoupled, and the inspiring network infrastructure is summarized. Consequently, carriers and enterprises achieve extraordinary automation, programmability, and network control, allowing them to create extremely adaptable, expansible networks that adjust instantly to ever-growing business requirements.

The ONF is an un-remunerative sector range that is leading the development of SDN, and establishing certain SDN infrastructure components like Open Flow Standardized protocol. Open Flow sets up interaction between the data and control planes of backed network equipment. Open Flow is the first typical interface considered especially for SDN, offering chapped traffic control, high-performance, all through a number of supplier’s network equipment.
Software Defined Networking [SDN] is a new and improved form of network architecture. In conventional methods of network architecture, network control was restricted to different network devices. But in SDN, the network control is separated from forwarding techniques. It has been made programmable in the form of accessible computing devices. Thus, various applications and network services can now access the underlying infrastructure. SDN enables the network to be treated as a logical or virtual entity.

The Data Plane: This is the last plane present in the structure. It’s main application is for forwarding the data packets. The Data Plane is also known as the Infrastructure Layer.

The Forwarding Plane: The Forwarding Network Elements make up the Forwarding Plane and is responsible for data forwarding, monitoring of local information and for gathering statistics.

The Control Plane: The Control Plane presents above the Data Plane and the Forwarding Plane and is responsible for programming and management of the Forwarding Plane. More specifically, it takes the information delivered by means of the Forwarding Plane and uses it to define network operation and for routing functions.

Fig. 1: Structure of SDN

2. LITERATURE SURVEY

If any link failure occurred in the traditional network. It’s not control the Particular Traffic. So it’s complicated to manage the Network. But the Software defined network can be able to manage the Traffic through the software programming. The SDN give the solution for traffic Control. Control the traffic through the control plane and data plane to overcome the problem to use one protocol is called open flow. It’s choose the correct path of the transmission data. Sometime the IOT system is merging to the SDN. So that easy to handle the traffic because it’s have connected to IOT so it have sufficient capacity control. The data arising from the network. It’s have the separating Method. Divide the one IOT network into small parts and each one is have different control. The controllers are control to the divided parts. Simulations.(Example: estinet and mininet) estinet is not sufficient working compare to mininet. But the mininet throughout the command working functions, so we use matlab functions. The SDN architecture have some components are SDN data path, SDN controller, SDN applications are present here. The SDN path is a logical network device that exposes visibility and uncontested control over it advertised forwarding and data processing capabilities. Then the SDN north bound interface is provide abstract the network views and its enable direct expression of network behaviors’ and requirements [1].

The heavy hitter detection helps to detect the traffic from networks. And improve the heavy hitter detection using the forwarding table entry methods are used. Switches are Identify the heavy hitters. Each switches gives the solution for reduce the heavy hitter to central server. The central server work is response to the switches identifying and detecting the heavy hitters. The heavy hitter detection work based on two inputs. That is open flow and traffic status. The traffic status is used to analyze the run time status in network. Then open flow is used to avoid the input debug to controller.
If any heavy hitter is found and then analyzes the heavy hitter is real (or) not. Then compare to the forwarding table entry (FTE) if it’s matched. And find out the heavy hitter and analyze the status. If new flow entry is added to the FTE. It’s not affected. The elephant flows are choose this method. If the elephant flow size is large to divide that flow or change the packet size. The small size of packets not make the traffic in that elephant flow. So the traffic is reduced in that particular flow. So the network flow will be good in SDN. The heavy hitter detection is represent in units in seconds [2].

A. The heavy hitter detection formula:

\[ T_{HH} = T_{comm} + T_{switch} + T_{controller} \]

The \( T_{comm} \) It’s denotes communication time between controllers & switches. Then \( T_{switch} \) is representing the querying time for traffic statistics. \( T_{controller} \) denotes the time to process traffic status at the controller side.

B. Thresholding:

\[ T^*_i = \begin{cases} s_t - T, & s_t \geq T \\ 0, & s_t > T \end{cases} \]

i is the index of the dataset. All the follows above threshold value t are considered as potential elephant flows. We have also using heavy hitter detection in software defined networks.

In this paper how will occurred the traffic in the SDN And what are the rules are present in the flows. The flows are using to reduce the traffic. The rules are provided to switches depends upon the incoming flow on the data plane. The new flow sizes are installed in devices of the topology. Then the unwanted rules are must be removed. The limited flow table size reduced by ternary content addressable memory and content addressable memory.

The packet length is depends upon the packet size and flow size. The packet size is calculated by packet size and packet arrival time. Only a specific packet travel can be done in the flows. The number of packets arrival the form of queue. To enter in to the switch the switches are have rules. Then flow table also here the flow table is compare to the packets depends upon packet size. Like these size of packets send this flow and then the packets are travel to the flow according to the rules. This is function of the software defined networking model [3].

In this paper said about traffic in SDN and reduce the traffic and management the traffic. The traffic measurement is help to analyze the real time network traffic. Reduced the traffic used to some existing methods that are load balancing energy saving scheduling. SDN have some traffic analyzing method, one is traffic analyze prediction and another one is network parameter prediction these all are traffic measurement related. Traffic load balancing and Qos scheduling, energy saving A literature survey on software defined network scheduling these are traffic management .The packet traffic is predicted by packet counter statistics. It’s mean analyze the network congestion and network efficiency based on traffic. The traffic measuring systems are three methods one is network topology parameter and another one is network performance parameter. The parameter measurement in the SDN switches kept the link layer discovery protocol from the controller. Then the switches are transmit the packets to the another switches. Then these switches are call the help to transmitting the packets with correct rule the switches have the flow table but not have that particular packet rule, so need help for packet transmitting. The Qos (guarantee scheduling) resources are to provide this system for traffic management .the SDN provides open flow interface to support the network traffic scheduling strategies [4].
3. EXISTING METHOD

Limited Flow Table Size highly influences the characteristics of the SOFTWARE DEFINED networks. The Table Flow can be controlled by providing each rule to the flow (IDEL Time and Hard-out Time). The controller of the SDN controls the rule to be performed in the particular flow. The main responsibilities of the controller is to assign the rule for the particular flow and govern the packets which match the rule, and sends the packet to that particular flow. There are many methods (Algorithms) which governs the table occupancy of the SDN switches. The main algorithms used are I.) Elephant Flows II ) Mice Flows.

A. Elephant Flows (a brief survey):

Elephant Flows are often called as the Heavy Hitters. Heavy Hitter Detection is used to identify the flows which have more traffic in it. These flows with more traffic are often called as Elephant Flows. In existing methods they choose this algorithm for identifying the flow with more Hit ratio (i.e. Number of promoted packets to the total number packets). By minimizing the hit ratio the average table occupancy can be reduced. The alternate method used is the identification of the small flows. SDN consists of more small flows. It is easier and more efficient way of reducing the table occupancy in SDN switch. Heavy Hitters can be identified in two ways they are I.)Size based Heavy Hitter Detection II.)Rate based Heavy Hitter Detection. In Size-Based, the heavy hitters are identified by introducing maximum threshold value (i.e. size value). Each flow is assigned with a maximum size and if a flow exceeds the maximum threshold point it is considered as a heavy hitter. (For example. 20 packets, 1000 packets, etc.). In Rate Based Heavy Hitter Detection, the maximum rate (i.e. Threshold point) is fixed. And when a flow exceeds this rate it is considered as a heavy hitter.

The Flow table occupancy is also controlled by using the average table flow occupancy and the 99th percentile table flow occupancy. The Maximum table flow occupancy cannot be used because the table may overflow and the major concern of SDN is the Limited Flow occupancy. The problem overruled by optimizing the functions. The cost function is chosen (i.e. Hit ratio) and the objective function is chosen as the (average table flow occupancy). The existing methods used in reduction of table flow are the I) Expedited Eviction and II.) Delayed installation. The expected eviction is used in TCP flows. It uses the RST/FIN Flag header for identifying the hitter packets and evicting it when a new flow comes into the switch. The Delayed installation is used in non-TCP flows. It waits for multiple packets and then it installs the packets into the flow. The INTER-arrival time of the packets also play a major role in the table occupancy. The inter-arrival time depends on the last packet which was promoted. The inter-arrival time is represented as inter(x), where x is the packet promoted. Time(x) is the time at which the packet is promoted. The previous packet promoted is denoted as prev(x). The inter-arrival time is calculated using the formula,

\[
\text{Inter}(x) = \text{time}(x) - \text{time}(\text{prev}(x))
\]

The existing method (Heavy Hitter detection) reduces the table occupancy from 16% to 62% with less than 1.5% hit ratio.

In this paper they state that the average table occupancy is governed based on the 99th percentile average table occupancy. It mainly targets on the last part of the flow table. Knapsack solution is used in order to reduce the occupancy. The hit ratio is collectively taken as the knapsack solution. The traffic analysis is performed using the zip functions. The flow table of SDN depends upon,

1. Number of packets promoted.
2. Number of packets missed.
3. The time taken for the packet to choose the path.
Total number of active times taken by the promoted packets. The matrices which are required for table occupancy are, 1. Traffic length, 2. Traffic bandwidth, 3. Hit ratio, 4. Packet arrival rate (inter-arrival rate), 5. Idle time.

The current open flow implementations use the idle-time mechanism to calculate the average timeout. In this open flow implementation, it identifies and evicts the inactive times of the packets present in the flow and this mechanism is known as the static idle timeout.

EXPEDITED EVICTION: In expedited eviction, the packet after the FIN/RST flag may not always be the last packet. Thus the packet after the FIN/RST flag header is considered as the last packet and the rules after that flow will be evicted. This method is mainly used in TCP flows.

Delayed Installation: In non-TCP, it is very difficult or it is merely not possible to predict when the flow is going to be terminated. Thus delayed installation concepts are used. A new rule will be added when the first flow is detected and the new rule will be waiting until the last flow is evicted.

**Fig. 2: Flow diagram**

B. Procedural flow of existing approach

**Fig. 3: Procedural flow of existing approach**

4. PROPOSED METHOD

A. Fuzzy based method for traffic control:

Our proposed method is based on the fuzzy logic controller. Fuzzy controller is used to control the traffic which arises due to the congestion occurred in the flow. The control plane of the SDN is unproblematic to introduce any functionality into the network. The functionalities are added with a very low line of code.
The fuzzy controller uses a control mechanism known as fuzzy logic control system (FLCS). The FLCS consists of two fuzzy systems: 1.) Label switched path setup system (Lss) 2.) Traffic splitting system (TSS). The status of the delay and load is controlled by Lss. The Link capacity and Utilization Rate are managed by TSS. This maintains the congestion free path. By using this method the utility rate is increased to more than 95%. The main advantage of this method is that it splits the traffic for different paths. It allows reduces the delay below than 2.6 seconds. A fuzzy set is any set that allows its members to have different grades of membership (membership function) in the interval [0,1].

DNN and fuzzy logic is used in this method. DNN is a machine learning technique. DNN plays a major role in neural and SDN networks. Fuzzy logic is used in SDN in order to split the flow size (number of packets) in a particular flow if any congestion occurs in the path. These features gives a numerical description of empirical and linguistic skills. The proposed method combines the characteristics of DNN and fuzzy logic. The network using this fuzzy logic and DNN implementation has two layers: 1.) Tied convolution 2.) Max pooling.

Tied convolution: Tied convolution is used to determine that whether the two packets are of same size. It also provides the relationship between the two different packets in a flow. Fuzzy logic is used to determine the flow which has more traffic and splits it into different flows and sends it to the flow table. Thus by using this method traffic and congestion in a flow is reduced.

The performance matrices used in our proposed method are 1.) Traffic length 2.) Traffic 3.) Packet arrival rate 4.) Idle timeout 5.) Hit ratio

Fig. 4 : Delay estimation (heavy hitter detection)
Fuzzy logic models, called fuzzy inference systems, consist of a number of conditional "if-then" rules. For the designer who understands the system, these rules are easy to write, and as many rules as necessary can be supplied to describe the system adequately (although typically only a moderate number of rules are needed). The DNN is a machine learning technique based on statistical methods. The DNN can be used for classification or regression analysis and its aim is to find the most optimal hyper plane. Thanks to this feature, it is different from other neural networks. This neural network consist of the following distinct layers: two layers of tied convolution with max pooling, cross-input neighborhood differences, patch summary features, across-patch features, higher-order relationships, and finally a soft max function to yield the final estimate of whether the input images are of the same cell or not. Each of these layers is explained in the following subsections.
Fig. 8: Delay estimation (Fuzzy logic)

Fig. 9: Traffic estimation (fuzzy logic)

Fig. 10: Traffic length estimation (fuzzy logic)

TABLE 1: COMPARISON TABLE

| Parameter           | Existing Approach(Heavy Hitter) | Proposed Approach(Fuzzy Classifier) |
|---------------------|---------------------------------|-------------------------------------|
| Minimum error       | 6.9795                          | 6.9695                              |
| Delay               | 334.01                          | 328.7                               |
| Average density out | 50.4302                         | 49.736                              |
5. CONCLUSION

The improvement in SDN systems deals the traffic congestion in congested areas. Various methodologies has been applied in reducing the traffic congestion problem. Some of these include network applications, supporting the network architecture, parking enforcement, fuel draining, expansion of already available computer networks, elimination of traffic and some other metrics. Non-linearity and traffic principles of the traffic movement and the high cost associated with the expansion of the existing infrastructure of computer networks and other related problems, the traditional methods have become very unusual.

The Table occupancy and the packet loss is reduced effectively by using fuzzy logic. The proposed method has successfully detected the flows which are large (in terms traffic and congestion) and it reduced the table occupancy by applying the functions and different thresholds to reduce packet loss in SDN switch. The proposed approach reduces the average traffic and delay in the network. The proposed method can be extended to extract still a stronger network. Future work on this project will include the maximizing the table flow size and by reducing the number of packets entering the flow path.

6. REFERENCE

[1]. Sabeeh A. et al. “A Hybrid Intelligent Approach for Optimising Software Defined Networks Performance”, International conference on Information Communication and Management, 2016

[2]. K.-C. Lan and J. Heidemann, “On the correlation of internet flow characteristics,” Information Sciences Institute, University of Southern California, Tech. Rep. ISI-TR-574, July 2003

[3]. J. Wan, D. Zhang, Y. Sun, K. Lin, C. Zou, and H. Cai, ‘‘VCMIA: A novel architecture for integrating vehicular cyber-physical systems and mobile cloud computing,’’ Mobile Network. Appl., vol. 19, no. 2, pp. 153–160,2014

[4]. J. Wan, C. Zou, K. Zhou, R. Lu, and D. Li, ‘‘IoT sensing framework with inter-cloud computing capability in vehicular networking,’’ Electron. Commerce Res., vol. 14, no. 3, pp. 389–416, Nov. 2014.
[5]. C. Estan and G. Varghese, New Directions in Traffic Measurement and Accounting, in ACM SIGCOMM, Pittsburgh, Pennsylvania, USA, August 2002, pp. 323-336

[6]. M. Jarschel, T. Zinner, T. Hossfeld, P. Tran-Gia, and W. Kellerer, “Interfaces, attributes, and use cases: A compass for SDN,” Communications Magazine, IEEE, vol. 52, no. 6, pp. 210–217, June 2014.

[7]. Yang J., et al. “Coding Open-flow: Enable network coding in SDN networks”, International journal of Computer Networks and Communications, Vol. 7, No 5, September 2015

[8]. K.-C. Lan and J. Heidemann, “On the correlation of internet flow characteristics,” Information Sciences Institute, University of Southern California, Tech. Rep. ISI-TR-574, July 2003.

[9]. F. Yu, Z. Chen, Y. Diao, T. V. Lakshman, and R. H. Katz, “Fast and memory-efficient regular expression matching for deep packet inspection,” in Proc. of the ACM/IEEE Symposium on Architecture for Networking and Communications Systems (ANCS), San Jose, California, USA, 4-5 Dec 2006, pp. 93–102.

[10]. Sulyukova, “Analysis of Low power and reliable XOR-XNOR circuit for high Speed Applications”, Journal of VLSI Circuits And Systems 1 (01), 23-26, 2019.

[11]. Z ZAIN, “High Speed And Lowpower Gdi Based Full Adder”, Journal of VLSI Circuits And Systems, 1 (01), 5-9, 2019