Congestion Recovery Policy for Fast Failover Links in Openflow Switches in Software Defined Networks

[1]Reecha Sood, [2]Dr.Sandeep Singh Kang
[1]PhD Scholar, [2] Professor
[12]Department of Computer Science and Engineering, Chandigarh University, Gharuan
Email: reecha.coecse@cgc.edu.in, sandeepkang.cse@cumail.in

Abstract

SDN Networks enables a new kind of networking in which users pay less and avail lot of services in return. The software system problem is incompatible with TCP related services as it also has the responsibility of performing many important tasks such as sending data etc. The purpose of this paper is to explore restoration process, gaining a conversation path when there is a failure in the software defined network.SDN based Applications can interact with open daylight via northbound APIs, informing the network of application specific and dynamic traffic flow requirements. In the proposed work, each process has been re-assigned to the local route to avoid congestion on the way. We first make the network's infrastructure hard to deal with abnormal events, then we have developed an effective design method for managing data and implementing a congestion avoidance policy. The approach enables administration of remote device via a central controller, reducing failure recovery routes and minimizing the need to manually configure traditional routers in branch locations.

Keywords: congestion avoidance; slow start; open daylight; SDN.

I. INTRODUCTION

Software Defined Networking (SDN) has attracted a lot of attention in the first place because it addresses a lack of functionality in the Legacy Networking[1]. SDN also integrates network architectures into the cloud and provide services to other technologies. SDN makes the network easy and fast, moving the control plane out of the power of the data plane enables both planes to work independently, thus providing basic services.

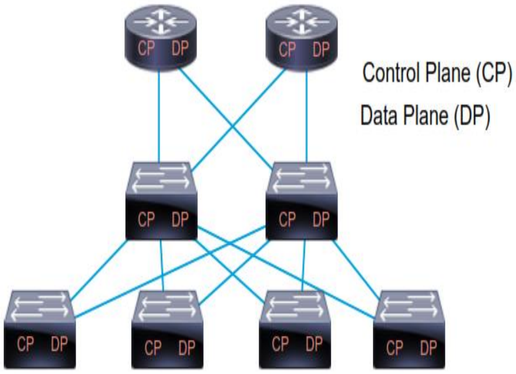


Figure.1. Control Plane and Data plane together in Traditional networks [10]

Software Defined Networking can manage services through the Network Manager. A summary of the low-level capabilities of the system, deciding on a system that decides where the traffic is sent (the control plane) to the underlying system that sends the traffic to the selected destination (data plane).The network manager performs the communication of the data plane via the control plane. Packets are sent by the network with the help of Openflow (OF) APIs[2].

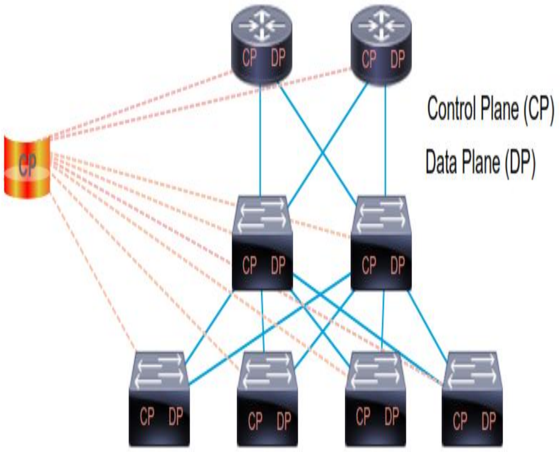


Figure.2. Control Plane and Data plane together in SDN 'purist'[10]

API use the control key to set paths, makes use of flow options to manage network traffic based on defined match rules that are (static or dynamic) matched to the related function which are also taken care by the OF protocol. Openflow protocol also investigates packet header and provides specifications (such as forwarding to certain ports, modifying content or captions, or drop packets).

Cloud-based technology generates a large amount of distributed traffic. These different applications can be difficult or complicated to handle in some types of situations[3].Therefore, too much pressure on network resources can be concentrated between one and many patterns to drive the network traffic towards data center. More importantly, in most data centers, more than one communication is implemented by Unicast, such as the Transmission Control Protocol, TCP, which is disabled. Because they generate a lot of reactive traffic, which not only ruins the network resources but also reduces the performance of the applications.

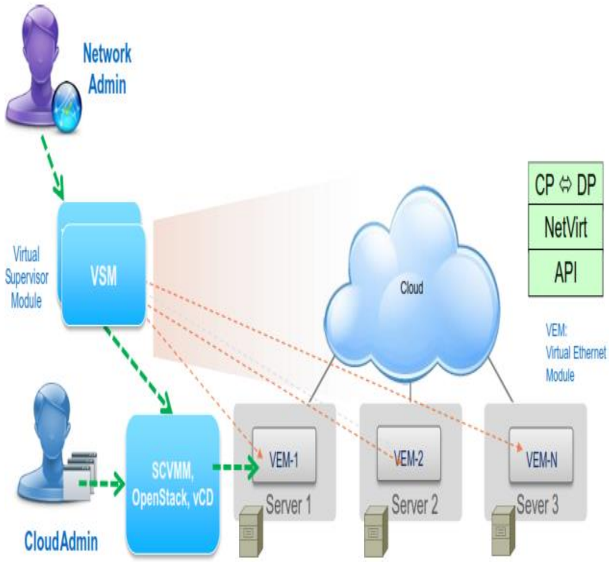


Figure. 3. Cloud-SDN hybrid Approach[16]

SDN deals with above situations and provides unique framework to handle the complicated problems in a simpler way. Moreover, the legacy networks suffers from congestion related problem. Now, the congested network is sorted through different mechanisms which are suggested by Transmission control Protocol. The SDN will pipeline with TCP mechanism to inform controller when any congestion occurs.

II. SYSTEMATIC REVIEW METHODS

TCP joint and split approach [4] places the TCP session between the client and the server machine. These machines are synchronized together and contain state information. The TCP option enables connection between the setup time and the client-to-join flow apart, and then split point-to-server portion of the flow during the SDN controller which preempts the TCP session. The authors have developed a framework that collects data from a cellular based system. The main system is linked to multiple base stations that share the common TCP path. The important task of the framework is to synchronize the entire system so that information is shared and easily accessed through split accesses. Using joint and split approach, the same type of functionality is provided to the system's control plane that remains in the main system. The proposed framework also follows the Linked-ACK based concept, maintaining the flow along the path from the server-based system to the client-based system. A client-based system is only available when the server receives the ACK packet. This new concept maintains end-to-end semantics. In addition, Linked ACK is managing the buffer data proportional to the sender side congestion window size. This way, it protects the buffer from unusual overflow in the Joint and Split proxy nodes.

TCP incast[5] is the problem created, when large number of users access the data at the same time. These users are connected to switch device, due to which they are quickly responded by the device.As a resultant some losses of data occur because the device is unable to cope with multiple senders at the same time. The TCP incast problems are timeout, delay and low latency which may trigger the performance losses in large organizations. To manage these problems, priority driven congestion control algorithm (PTCP) is proposed by the researchers. PTCP address the incast problem and solves the timeout, delay and latency factors. Each flow size is met with the deadline; the deadline can be obtained from the beginning of the data. The basic principle of this approach is to maintain flow according to tight deadlines. The higher priority is given to the flow of tight deadline over non-deadline flows. Those flows that miss the deadline are dropped and are expected to complete the deadline of second cycle. In addition, the receiver size window setting is also implemented in the PTCP. The receiver window size depend on the Explicit Congestion Notification (ECN)[6], if the accurate feedback is attained by the sender machine, the obtained value of queue delay is minimum and performance of the network is high.

In order to adapt diverse application and networks, together with various requirements of the customers to be fulfilled, Multipath TCP (MPTCP) protocol is used for this purpose. MPTCP protocol is the extension of TCP protocol in which the authors have investigated and measured the performance of the network[7]. It was studied that the performances of the network fall down due to packet losses or congestion in the networks. Slow start Algorithm is the most impressive method to address congestion related problems but when multiple users share the same application through multi paths then the buffer overflow problem arises in the gateway. Gentle slow start scheme (GSAM) for MPTCP protocol has been used to solve the buffer overflow and congestion related problems. GSAM include short flows by maintaining the threshold value corresponding to each path and to avoid router overflow.

In the wifi-offloading paradigm based SDN [8], the authors have solved the two fundamental problems namely (1) to verify the data whether it is received by the receiving station or not and (2)feedback system received by the Access point (AP) through both machines. In the wifi-offloading process each channel has to complete the flow twice before being connected to Access Points.

III. LITERATURE SURVEY

Ahmed A. Alabdel Abass, Mohammad Hajimir sadeghi, Narayan B. Mandayam, Zoran Gajic (2016) [13] discussed the distributed denial of service attack with EGT i.e. Evolutionary Game Theory. In this theory it is totally depended on the user whether they transmit the data by measuring the transmission probability method. This method has been implemented on Physical Layer where signal strength calculation was done in this research paper. The authors called the attacker as 'Jammers' that launches the attack on the common users and makes accessible the payload of the data. Here, it is assumed that there are m users and n jammers that correspondingly attack on the network and get benefit from the common users. This paper calculates the Signal to inference plus noise ratio (SINR) and if the value of SINR is lower than it assumes that attacks trigger on the network.

Kashif Saghar, Hunaina Farid, David Kendall, Ahmed Bouridane (2016) [14] described the Denial of Service (DoS) that erupted at the wireless node of the network. As a result, other wireless node of the same and different group of the network were affected. This attack was erupted by the group of attacker or single attacker due to accessing the single channel or compromising the nodes in the network. The increasing number of attacks does not only because single network damages but also adds to the malfunctioning or viruses in the network. However, due to weak policies, methodology and other low security devices shrink the security of the network.

A.Kannammal, S.Sujith Roy (2016) [15] has rapped that attacks may be active or passive. Passive attacks have stolen some useful information that was circulated in between the two node and we assumed that the information was very valuable for both parties. While the Active attack carry on the same network but this attack is less dangerous than passive attack. This attack disrupts the data during normal operations. The common attacks were performing on the OSI model - Physical Layer, Data Link Layer, Network Layer, Transport Layer and Application Layer. There were also many security solutions but not triggering due to insufficient resources.

H. Xu et.al. (2017) [9] has designed novel switching technique that collaborate both traditional switching technique and SDN technique for resolving scalability and performance based factors. The scalability problem had solved through conventional principle that was not determined as per flow based routing but also implemented destination based address. The same path has been allocated to all the nodes when the destination address becomes unique. Another principle is that the network would be decentralized so that the topology is not fixed on centralized node. The combination of two techniques solves numerous issues like bottleneck, single point failure, congestion, Ethernet learning based algorithm and many more. The SDN network has been carefully designed so that all traditional switching qualities have been implemented on the hybrid technique. The authors show that hybrid approaches have many benefits than traditional approaches but the work is not based on forwarding rules of the routing and route performance could not match with traditional approach.

S. V Galich, M. S. Deogenov, and E. S. Semenov (2017) [10] Classification and analyzing the types of delay in software defined networks which are measured through open Day Light controller and mininet emulator. Address resolution protocol (ARP) collecting the hardware details of the networking devices that was implemented in the algorithm. The algorithm investigated the traffic packet like header, length of the packet and fragmentation. On the basis of the value it provides the services to various connected devices and controls the latency factor. If the packet latency is more than existing approach then controller re-examines all the factors which were earlier calculated otherwise suitable recommendation and procedures are adopted to improve the performance. Some of the recommendations are given below: switch latency depends on switch matrix, size of the buffer and packet queue size of the physical interface. Controller latency depended on the hardware and application code. The application depends on hardware which was used and operated on the platform.

J. H. Cox et al. (2017) [11] has surveyed the software defined networks and explored new technologies where the work would be starting by new researchers. The SDN research has been divided into number of technologies like deployment in industries, government sectors and campus area networks. Furthermore, there are a number of benefits of this technology such as you don't pay more for any hardware—if it is not necessary. The SDN is application view only so anyone can have access to the application and to the hardware based open switches to send and store data. However, the challenges are integration of traditional switching and SDN switching, routing operations and some security issues. Another important research opportunity is to remove barriers between network operators and organizations. This can be extended via simulators but it is not an easy task to simulate such kind of environment that supports application and hardware. At last, research scopes towards hybrid SDN solutions that partially covers the traditional and next generation infrastructure like IPV4, IPV6, cloud and IoT Technologies.

Y. Zhao, P. Zhang, Y. Wang, and Y. Jin (2017) [12] discussed SERVE and SDN rule verification framework. These two approaches has automatically recognized the problems of SDN data plane. The number of rules written in SERVE based model generates number of probes. These probes are injected into the network pipeline throughout-band channel that constructed multi-rooted tree. When processing the probes in the network which collects network configuration, behavior of the network, topology change of the data plane after verification and comparison process starts in the control plane. This SERVE rule determines the location of the data plane and then informs the controller. The controller has stored the information and that information is updated to the routing tables in timely manner. Another proposed method is SMRT refers to stateful multi rooted tree that captures the traffic through pipeline and compares the two values (i.e. state information and modifies header field). SMRT makes it possible to track the traditional value, stores them along with packet on the path [4].

In this section, we have also summarized some of the protocols and related algorithms that have been used in the proposed framework to improve the software defined networking.

A. Slow Start:

In the SDN networks, the transmission starts with slow start during data plane in the presence of controller. The slow start maintains the congestion window of sender and receiving machine. Increase the crowd window through each receipt received. The size of the congestion window only increases when no approvals or segments are lost [16]. This access indicates the performance of the TCP protocol, when the communication is restarted after the handoff process; the slow start process is initiated. During the handoff process, the congestion window size shrinks, decreasing by one. In wireless instances, handoff problems are common and therefore require long waiting by the sending machine. To accelerate the long waiting period, 3 duplicate ACK process are started. In this process, there is a 3-copy ACK waiting for the presenter, if the ACK is not received from the recipient side then the retransmission process is initiated.

B. Congestion Avoidance:

Upon reaching the threshold value of the congestion window, the SDN system enters the congestion window position. In this case, the size of the cwd is set to half of the cwd window. In each part transmitted, the sender handles the timer[17]. Assume that the maximum window size is W and the minimum window size is W / 2 (constant position). The total change in window size is w / 2 throughout the whole process. If the timer expires before ACK is received, the system will start to slow start state. The timer is updated according to Round Trip Time (RTT). Suppose window open for one packet at a time so that the total time to submit the data is RTT * W / 2. At each process, the timer may be updated based on the RTT; again this can be estimated on the basis of ACK received. The main challenge in our proposed framework is to control the growth of the congestion window while this is very difficult to tune with timer. The initial subflow starts off as a regular TCP, which will be seen in the next section, doubling the window size of each cycle upon receiving a new ACE. If the flow is not maintained during one or two cycles, the crowd window drops and the algorithm goes directly to the congestion window position. The purpose of this paper is to make SDN a "best effort", capable of handling every effort and maintaining the quality of service.congestion avoidance phase[16]. The state enables high congestion or link failures over time. Congestion can also occur in multicast traffic, and to avoid imbalanced network loads, routing protocols can be planned to work according to the fast retransmit state.

B Fast Recovery and Fast Retransmit:

Upon reaching the threshold value of the congestion window, the SDN system enters the congestion window position. In this case, the size of the cwd is set to half of the cwd window. In each part transmitted, the sender handles the timer[17]. Assume that the maximum window size is W and the minimum window size is W / 2 (constant position). The total change in window size is w / 2 throughout the whole process. If the timer expires before ACK is received, the system will start to slow start state. The timer is updated according to Round Trip Time (RTT). Suppose window open for one packet at a time so that the total time to submit the data is RTT * W / 2. At each process, the timer may be updated based on the RTT; again this can be estimated on the basis of ACK received. The main challenge in our proposed framework is to control the growth of the congestion window while this is very difficult to tune with timer. The initial subflow starts off as a regular TCP, which will be seen in the next section, doubling the window size of each cycle upon receiving a new ACE. If the flow is not maintained during one or two cycles, the crowd window drops and the algorithm goes directly to the congestion window position. The purpose of this paper is to make SDN a "best effort", capable of handling every effort and maintaining the quality of service.

C. Fast Recovery and Fast Retransmit:

In this case, the congestion window (cwnd) is updated using the same algorithm that is used in the congestion avoidance phase[16]. The state enables high congestion or link failures over time. Congestion can also occur in multicast traffic, and to avoid imbalanced network loads, routing protocols can be planned to work according to the fast retransmit state.

When the network is congested, the normal TCP focuses on the fast retransmit and the faster recovery based solution[5]. The goal of our algorithm is to reduce the value retransmit timeout and faster re-broadcast so that it can support multiple demand connections. The use of TCP in an SDN-based network for training for congestion based problems as UDP sessions receive longer sessions. If we take an example of an SDN network, the switch receives a broadcast frame from the data center; it sends the frame out to every port except ingress port where the frame was found Broadcast sometimes necessary to locate other devices in the network and give this information to the controller, but they also reduce the network traffic. Network bandwidth is directly propagating the broadcast traffic. Network bandwidth promotes live broadcast traffic. Many broadcasts can result from congestion, which slows down SDN's performance. If the source determines that the TCP component is not being recognized in a timely manner, the Fast Retransmit State is activated.

D. Selective Acknowledgment:

The recipient usually sends an acknowledgment after receiving each of the information. The segments are received before being considered. From the actual message received by the recipient before it understands, the data in these sections is reordered in the original format. The sequence number is assigned to the header of each segment to achieve the goal. Sequence number represents the first data byte of the TCP partition.

Data Byte Tracking enables each segment to be uniquely identifiable and recognizable.TCP complies with the cumulative acknowledgment Scheme, which is used in the initial phase. It does not provide complete information to the user as it recovers packet loss in each cycle within a transmission window. Each acquisition proposed by SACK [18]proposed each acknowledgment contain up to three noncontiguous blocks of data, each block is assigned with a starting sequence number and ending sequence number.

This above-mentioned problem can be handled in a systematic way through our method. The method ensures that the only logical path between all the destinations that causes a loop is to block unnecessary paths.

IV. RESEARCH PLANNING

In this section, we implement slow start algorithm in software defined networking (SDN). We assume that packet loss occurred in random fashion and TCP adjust the window size in each short flow. We are taking following assumptions in our experimental scenario:

1. Round Trip Time (RTT) is constant.
2. Packet loss probability is ‘p’ which is also taken as constant.
3. Sender transmits the data in each cycle and w is the congestion window maintained in gateway and destination machine.
4. Retransmit timeout is not considered in the normal TCP based scenario.

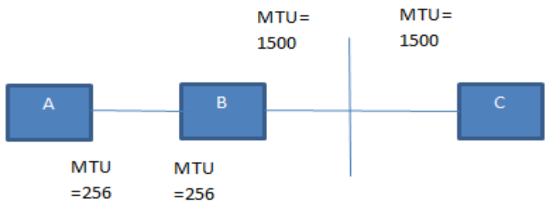


Figure.4. Slow Start Algorithm Approach in SDN[18]

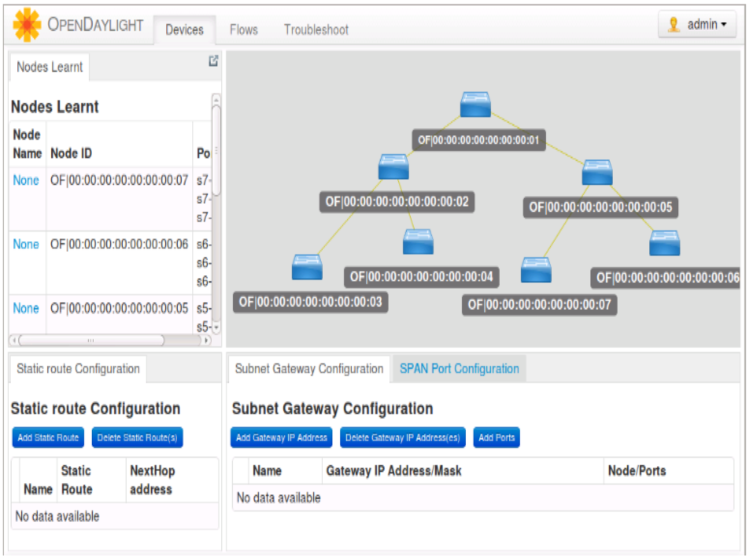
Maximum segment size (MSS) is the largest “chunk” of data that TCP will send to other end[39]. When a connection is established, each end can announce its MSS. In general, MSS have used 1024 bytes. The resulting IP datagram is normally 40 bytes larger: 20 bytes for the TCP Header and 20 bytes for the IP Header.If one end does not receive an MSS option from the other end, a default 536 bytes is assumed.The planning stage shows the criteria followed on basic parameters of software defined networks to improve the quality of the network. The following table 1provide the layout of basic planning of the network.

Table I. Quality Criteria

Parameter	Criteria	References
Inclusion	Join and split approach, TCP, MPTCP, slow start	[4][5][7]
Exclusion	Flow monitoring scheme	[40]
Quality	Throughput performance of TCP flow, TCP delay in wireless environment, Multipath TCP throughput; evaluation of traffic distribution, ACK delay	[4], [5]

V. PERFORMANCE EVALUATION

We investigated proposed system through open day light controller, the controller especially used in software defined networking. In the simulation network, seven switches are employed in the large network. A virtual local area network (VLAN) can be created on layer-2 switch to reduce the size of broadcasting and act like a Layer-3 switch.VLANs incorporated in network design making it easier for SDN network to support the goal of a proposed scenario. Multiple IP subnets can exist in the proposed system that can span multiple physical LAN segments. VLANs trunking also implemented in our scenario which extends VLANs across an entire network. Trunking allow all VLAN traffic to propagate between switches which are in same VLAN.


OPENDAYLIGHT
Devices
Flows
Troubleshoot
admin
Nodes Learnt
Static route Configuration
Subnet Gateway Configuration
SPAN Port Configuration
Add Static Route
Delete Static Route(s)
Add Gateway IP Address
Delete Gateway IP Address(es)
Add Ports
OF|00:00:00:00:00:00:00:01
OF|00:00:00:00:00:00:00:02
OF|00:00:00:00:00:00:00:05
OF|00:00:00:00:00:00:00:04
OF|00:00:00:00:00:00:00:06
OF|00:00:00:00:00:00:00:03
OF|00:00:00:00:00:00:00:07

Nodes Learnt

Node Name	Node ID	Po
None	OF|00:00:00:00:00:00:00:07	s7- s7- s7-
None	OF|00:00:00:00:00:00:00:06	s6- s6- s6-
None	OF|00:00:00:00:00:00:00:05	s5- s5-

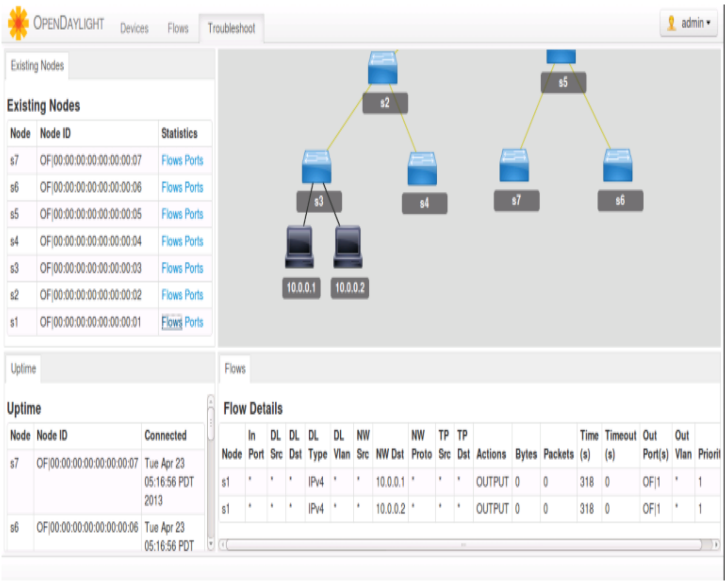
Static route Configuration

Name	Static Route	NextHop address
No data available		

Subnet Gateway Configuration

Name	Gateway IP Address/Mask	Node/Ports
No data available		

Figure. 5. USENET Topology[23]



Existing Nodes

Node	Node ID	Statistics
s7	OF|00:00:00:00:00:00:00:07	Flows Ports
s6	OF|00:00:00:00:00:00:00:06	Flows Ports
s5	OF|00:00:00:00:00:00:00:05	Flows Ports
s4	OF|00:00:00:00:00:00:00:04	Flows Ports
s3	OF|00:00:00:00:00:00:00:03	Flows Ports
s2	OF|00:00:00:00:00:00:00:02	Flows Ports
s1	OF|00:00:00:00:00:00:00:01	Flows Ports

Uptime

Node	Node ID	Connected
s7	OF|00:00:00:00:00:00:00:07	Tue Apr 23 05:16:56 PDT 2013
s6	OF|00:00:00:00:00:00:00:06	Tue Apr 23 05:16:56 PDT

Flow Details

Node	In Port	DL Src	DL Dst	DL Type	DL Vlan	NW Src	NW Dst	NW Proto	TP Src	TP Dst	Actions	Bytes	Packets	Time (s)	Timeout (s)	Out Port(s)	Out Vlan	Priorit
s1	*	*	*	IPv4	*	*	10.0.0.1	*	*	*	OUTPUT	0	0	318	0	OF|1	*	1
s1	*	*	*	IPv4	*	*	10.0.0.2	*	*	*	OUTPUT	0	0	318	0	OF|1	*	1

Figure.6. Topology Communication: host added in SDN network[23]

6.1 Results

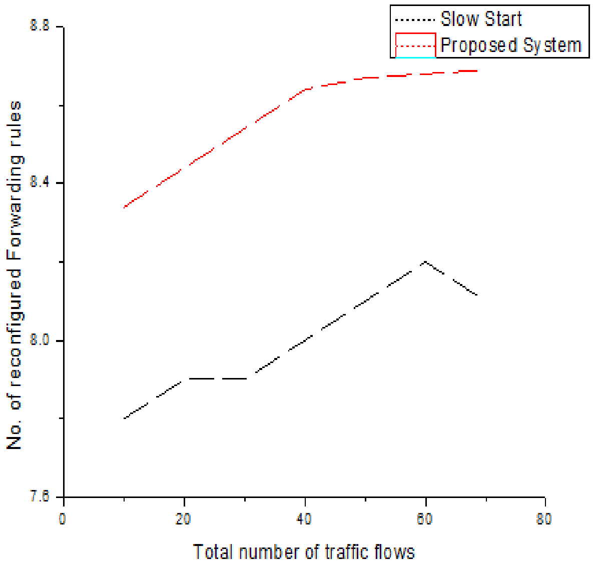


Figure. 7. Number of reconfigured forwarding rules

Figure 7 shows that the number of reconfigured forwarding rules after the failure of link. The results in the figure are average run of independent switches which are shown in the topology. We can see that forwarding rules are maintained by the controller, thus the proposed approach maintain the forwarding rules efficiently than the existing approach.

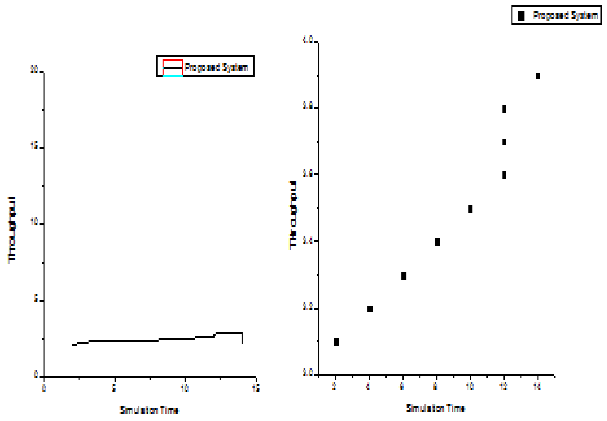


Figure. 8. Throughput of Proposed System

If the forwarding rule data increases than it is difficult to maintain the queue list because the queue list is maintained by the device buffer. Figure 8 shows the throughput of the proposed system that compare with existing system. We can see that throughput of existing system decreases because the queue size increases and we say that forwarding rule proportional to the size of queue length. When the queue length increases, at one time it drops the packets and then throughput constantly decreases.

VI. SUMMARY AND FUTURE WORK

In this paper, we have developed a framework that is able to solve problems associated with software-defined networks. The functionality of the framework is simple which eliminates and redistributes information between overlay control protocols such as controllers and TCPs. All encapsulated flows look like the same flow between the edge devices. It is also reported that the encapsulated traffic can exceed the maximum MTU of the route. Comprehensive simulation results showed that compared to previous SDN access, our proposed approach avoids congestion in the network and provides local access faster than other aches methods.

In the future work will increase the level of security in the network as defined by the accessibility software. Increasing the level of security can lead to greater confidentiality and authentication within the system.

REFERENCES

[1] J. Costa-Requena *et al.*, “Software defined 5G mobile backhaul,” in *Proceedings of the 2014 1st International Conference on 5G for Ubiquitous Connectivity, 5GU 2014*, 2014, no. November, pp. 258–263.

[2] R. Riggio, M. K. Marina, J. Schulz-Zander, S. Kuklinski, and T. Rasheed, “Programming Abstractions for Software-Defined Wireless Networks,” *IEEE Trans. Netw. Serv. Manag.*, vol. 12, no. 2, pp. 146–162, 2015.

[3] S. M. Mousavi and M. St-Hilaire, “Early detection of DDoS attacks against SDN controllers,” *2015 Int. Conf. Comput. Netw. Commun. ICNC 2015*, pp. 77–81, 2015.

[4] W. Guo, V. Mahendran, and S. Radhakrishnan, “Join and spilt TCP for SDN networks: Architecture, implementation, and evaluation,” *Comput. Networks*, vol. 137, pp. 160–172, 2018.

[5] J. Zhuang, X. Jiang, G. Jin, J. Zhu, and H. Chen, “PTCP: A Priority-Driven Congestion Control Algorithm to Tame TCP Incast in Data Centers,” *IEEE Access*, vol. 7, pp. 38880–38889, 2019.

[6] S. Fowler, M. Eberhard, and K. Blow, “Implementing an adaptive TCP fairness while exploiting 802.11e over wireless mesh networks,” *Int. J. Pervasive Comput. Commun.*, vol. 5, no. 3, pp. 272–294, 2009.

[7] P. Dong, W. Yang, K. Xue, W. Tang, K. Gao, and J. Huang, “Tuning the Aggressive Slow-Start Behavior of MPTCP for Short Flows,” *IEEE Access*, vol. 7, pp. 6010–6024, 2019.

[8] I. V. S. Brito and G. B. Figueiredo, “Improving QoS and QoE through seamless handoff in Software-Defined IEEE 802.11 Mesh Networks,” *IEEE Commun. Lett.*, vol. 21, no. 11, pp. 2484–2487, 2017.

[9] H. Xu, “Scalable Software-Defined Networking through Hybrid Switching,” pp. 1–9, 2017.

[10] S. V Galich, M. S. Deogenov, and E. S. Semenov, “Control Traffic Parameters Analysis in Various Software-defined Networking Topologies,” 2017.

[11] J. H. Cox *et al.*, “Advancing Software-Defined Networks : A Survey,” vol. 99, pp. 1–40, 2017.

[12] Y. Zhao, P. Zhang, Y. Wang, and Y. Jin, “Troubleshooting Data Plane with Rule Verification in Software-Defined Networks,” vol. 14, no. 8, 2017.

[13] A. A. A. Abass, M. Hajimirsadeghi, N. B. Mandayam, and Z. Gajic, “Evolutionary game theoretic analysis of distributed denial of service attacks in a wireless network,” in *2016 50th Annual Conference on Information Systems and Sciences, CISS 2016*, 2016, pp. 36–41.

[14] D. Kendall, “Formal Specifications of Denial of Service Attacks in Wireless Sensor Networks.”

[15] A. Kannammal and S. S. Roy, “Survey on Secure Routing in Mobile Ad- hoc Networks,” 2016.

[16] K. Sohraby, D. Minoli, and T. Znati, *Wireless Sensor Networks*. 2007.

[17] Q. Gao, W. Tong, S. Kausar, L. Huang, C. Shen, and S. Zheng, “Congestion-aware multicast plug-in for an SDN network operating system,” *Comput. Networks*, vol. 125, pp. 53–63, 2017.

[18] H. Balakrishnan, V. Padmanabrhan, S. Sesrhan, and R. H. Katz, “A Comparison of Mecrhanisms for Improving TCP Performance over Wireless Links,” *IEEE/ACM Trans. Netw. %8 December %! A Comp. Mecrhanisms Improv. TCP Perform. over Wirel. Links.*, vol. 5, no. 6, pp. 756–769, 1997.

[19] H. Balakrishnan and V. N. Padmanabhan, “How network asymmetry affects TCP,” *IEEE Commun. Mag.*, vol. 39, no. 4, pp. 60–67, 2001.

[20] B. Francis, V. Narasimhan, A. Nayak, and I. Stojmenovic, “Techniques for enhancing TCP performance in wireless networks,” in *Proceedings - 32nd IEEE International Conference on Distributed Computing Systems Workshops, ICDCSW 2012*, 2012, pp. 222–230.

[21] Y. Li, S. Lei, X. You, H. Zhuang, and K. Sohraby, “Performance of TCP in intermittently connected wireless networks: Analysis and improvement,” in *GLOBECOM - IEEE Global Telecommunications Conference*, 2010, pp. 1–6.

[22] G. Buchholcz, A. Gricser, T. Ziegler, and T. Van Do, “Explicit Loss Notification to Improve TCP Performance over Wireless Networks,” *Springer-Verlag Berlin Heidelb.*, pp. 481–492, 2003.

[23] M. A. Alrshah, M. A. Al-Maqri, and M. Othman, “Elastic-TCP: Flexible Congestion Control Algorithm to Adapt for High-BDP Networks,” *IEEE Syst. J.*, vol. 13, no. 2, pp. 1336–1346, 2019.

[24] A. Srikanth, P. Varalakshmi, V. Somasundaram, and P. Ravichandiran, “Congestion Control Mechanism in Software Defined Networking by Traffic Rerouting,” *Proc. 2nd Int. Conf. Comput. Methodol. Commun. ICCMC 2018*, no. Iccmc, pp. 55–58, 2018.

[25] J. Zheng *et al.*, “Congestion-Free Rerouting of Multiple Flows in Timed SDNs,” *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 968–981, 2019.

[26] T. Zhu *et al.*, “A congestion-aware and robust multicast protocol in SDN-based data center networks,” *J. Netw. Comput. Appl.*, vol. 95, no. January, pp. 105–117, 2017.

[27] M. Prakash, A. Abdrabou, and W. Zhuang, “An Experimental Study on Multipath TCP Congestion Control with Heterogeneous Radio Access Technologies,” *IEEE Access*, vol. 7, pp. 25563–25574, 2019.

[28] J. Ye, L. Feng, Z. Xie, J. Huang, and X. Li, “Fine-grained congestion control for multipath TCP in data center networks,” *IEEE Access*, vol. 7, pp. 31782–31790, 2019.

[29] Z. Cheng, X. Zhang, Y. Li, S. Yu, and R. Lin, “Congestion-Aware Local Reroute for Fast Failure Recovery in Software-Defined Networks,” vol. 9, no. 11, pp. 934–944, 2017.

[30] Y. Lu and S. Zhu, “SDN-based TCP congestion control in data center networks,” *2015 IEEE 34th Int. Perform. Comput. Commun. Conf. IPCCC 2015*, 2016.

[31] B. B. Letswamotse, R. Malekian, C. Y. Chen, and K. M. Modieginyane, “Software defined wireless sensor networks and efficient congestion control,” *IET Networks*, vol. 7, no. 6, pp. 460–464, 2018.

[32] T. Hafeez, N. Ahmed, B. Ahmed, and A. W. Malik, “Detection and Mitigation of Congestion in SDN Enabled Data Center Networks: A Survey,” *IEEE Access*, vol. 6, pp. 1730–1740, 2017.

[33] Y. Hu, T. Peng, and L. Zhang, “Software-Defined Congestion Control Algorithm for IP Networks,” *Sci. Program.*, vol. 2017, 2017.

[34] C. Lin, K. Wang, and G. Deng, “A QoS-aware routing in SDN hybrid networks,” *Procedia Comput. Sci.*, vol. 110, pp. 242–249, 2017.

[35] 2019 Haines et al, “ECTCP:An explicit centralized congestion Avoidance for TCP in SDN based Data Center,” in *IEEE Symposium on Computers and Communications (ISCC)*, 2013, vol. 53, no. 9, pp. 1689–1699.

[36] L. M. Elguea and F. Martinez-Rios, “A New method to optimize BGP routes using SDN and reducing latency,” *Procedia Comput. Sci.*, vol. 135, pp. 163–169, 2018.

[37] Y. Cao, “Bifurcations in an Internet congestion control system with distributed delay,” *Appl. Math. Comput.*, vol. 347, pp. 54–63, 2019.

[38] R. Muñoz *et al.*, “Integration of IoT, Transport SDN, and Edge/Cloud Computing for Dynamic Distribution of IoT Analytics and Efficient Use of Network Resources,” *J. Light. Technol.*, vol. 36, no. 7, pp. 1420–1428, 2018.

[39] F. Granelli, D. Kliazovich, and N. L. S. da Fonseca, “Performance limitations of IEEE 802.11 networks and potential enhancements,” *Int. J. Comput. Res.*, vol. 14, pp. 85–108, 2007.

[40] C. N. Sminesh, E. G. M. Kanaga, and K. Ranjitha, “Flow monitoring scheme for reducing congestion and packet loss in software defined networks,” *2017 4th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2017*, pp. 1–5, 2017.

Authors:

R Sood is currently working as an assistant professor in CSE department at Chandigarh Group of Colleges-College of Engineering, Landran, Mohali and pursuing her Ph.D in CSE (Sofware Defined Networks) from Chandigarh University, Gharuan.

Dr. Kang is currently working as a Professor in the Department of Computer Science and Engg. at Chandigarh University, Gharuan.