IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 3, 2018, accepted February 5, 2018, date of publication February 28, 2018, date of current version March 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2808358

Leveraging the Big Data Produced by the Network to Take Intelligent Decisions on Flow Management

SHAFAAN KHALIQ BHATTI[1], M. IKRAM ULLAH LALI[2], BASIT SHAHZAD[3], FAROOQ JAVID[1], FAKHAR ULLAH MANGLA[1], AND MUHAMMAD RAMZAN[1]

[1]Department of Computer Science and IT, University of Sargodha, Sargodha 44100, Pakistan
[2]Department of Software Engineering, University of Gujrat, Gujrat City 50700, Pakistan
[3]Department of Engineering, National University of Modern Languages, Islamabad 44000, Pakistan

Corresponding author: M. Ikram Ullah Lali (ikramlali@gmail.com)

ABSTRACT Software-defined network (SDN) offers a very advantageous feature of programming the network at run time by decoupling the control plane from data plane to have centralized and better forwarding decisions. To attain the maximum throughput and relatively less latency within a large network, we have captured the real-time big data produced by the network. Big data is revolutionizing the modern computer science world to analyze the extremely large datasets to predict the future requirements. Therefore, it would be useful to embed intelligence in the systems. With the power of SDN, we programmed our network at run time using Ryu controller to dump the network traffic and engineer it accordingly. We proposed a methodology for taking intelligent decisions with traffic engineering in SDN. Results have shown that SDN leverages the big data to decrease the latency time by adding programs in the controller of SDN. An algorithm is presented to maximize the bandwidth utilization to possess higher throughput after shaping the traffic. It is found through the results that our proposed methodology of applying intelligence on traffic management in SDN outperforms those without intelligence decision making.

INDEX TERMS SDN, big data, Mininet, intelligent decisions, flow management.

I. INTRODUCTION

One of the major trends in the networking industry is software-defined networking (SDN) architecture, which provides architectural support for ''programming the network''. It uses a centralized controller that can remotely program multiple data planes through a standardized open interface. The resultant system architecture of SDN with separate control and data planes is an open, horizontally integrated architecture [1]. This condition is contrary to the closed, vertically integrated architecture, which is typical of traditional networking and widely perceived as inflexible. In these networks, the control and data planes are directly coupled and jointly controlled by networking vendors through proprietary implementation. The SDN proposal mainly aims to fix the rigidified architecture of the Internet, which makes introducing network innovations difficult. The major insight of SDNs is the programmability at the network level through an open programming interface rather than vendor-specific switch-level command line interfaces. The SDN architecture commits to revolutionize the networking using rapid innovations by promoting modularity, reuse sophisticated abstractions and open standards.

SDN separates the data plane of the network from the control plane and acts as a type of network architecture. SDN is the software logic for managing traffic through a single high-level program. SDN has many benefits, including centralized and decentralized control of various cross-vendor network elements, mainly data plane platforms with a shared abstraction layer of an application programming interface (API) for all SDN-facilitated equipment [2]. Moreover, SDN minimizes the complexity of network configuration and operations by automating a high-level configuration that is converted into a specific forwarding behavior of network elements. SDN also enables the easy deployment of new protocols and network services through high operation abstraction [3]. The augmented control granularity in SDN enables a per-flow definition with an elevated granularity policy level [4]. An SDN infrastructure can be adjusted through the control plane to a specific user application running on this approach by improving user experiences [5].

It is noteworthy from the above discussion that SDN have revolutionized the networks by adding flexibility in network management and traffic engineering. Furthermore, in recent years data trends have changed and we need special attention to the data traffic engineering in the new networks. Internet of things (IoT) has emerged as new big data producing devices at a large scale [6]. B*ig data*, which refers to the emerging capability of processing and mining massive amounts of data. Data processing is difficult using data warehousing and the conventional database technology because of variety, volume, and velocity, also known as the three V's of big data. A staggering amount of 2.5 exabyte per day is created with new data [7]. Considerable attention is required when using information in "big data" for analytics, the benefits of which include modeling and understanding of latent trends (*exploratory* and *descriptive analytics*) and *predictive analytics*, which aim to forecast future trends. The major challenge in big data analytics is finding the proverbial needle in the haystack that improves operations and services. Therefore, latent knowledge, interconnection, and structure lead to insightful decisions. Field tools for analytics, such as learning from a machine, time series analysis, statistical formulation, data warehousing, business intelligence, and mining of data besides processing of big data using expert methods, are exploring its application to different modern life spheres, such as education, online stores, telecommunications, government, and business [8].

SDN and big data are typically implemented and addressed individually, but the former with its growing popularity as a result of its inspiring features along with the latter now work together for future networking phenomena. SDN fully supports big data in transmission, manipulation, processing, and storage [1]. In turn, big data assists SDN in optimizing operations [7]. SDN is used in many big data applications for scheduling and process handling in cloud-enabled data centers [4]. Efficient data delivery and joint optimization of big data are implemented by SDN as shown in Figure 1. Network performance and utilization may be enhanced by using SDN for run-time network configuration in big data applications [5]. Big data protects SDN from security attacks in cross-layer designing and traffic engineering. It plays a vivacious role in the intra- and inter-cloud data center networks [9].

In the University environment, IoT is a major source of big data. It is contributing to give data services and it is becoming a vital element for communication, information sharing, and data generation. Internet based objects require significant attention to process these data into useful information. In a university domain, data produced by IoT is needed to handle with the novel ideas and management facilities of SDN to enhance the performance of the university [10]. The software-Defined Networking (SDN) handles the IoT generated data better than traditional networks by abstracting all the controls and management tasks

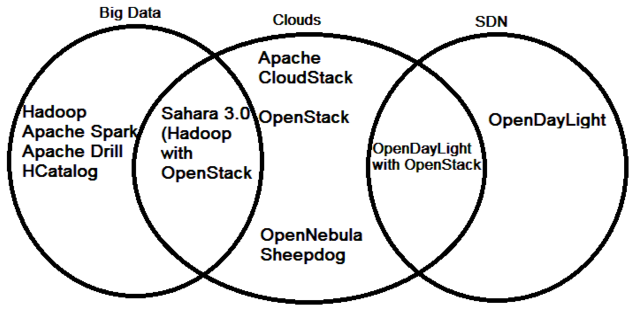

FIGURE 1. Relationship among different open source projects of Big Data & SDN.

from things in the IoT. SDN sets IoT and IoMT (Internet of Multimedia Things) inside a middleware software layer and this gives an efficient data handling approach in the University configurations [11]. SDN deployment on virtual machines (VMware, NSX) through an advanced network OS like Cumulus or Big Switch enables the computation to move from the core to the edge. Software-Defined Networking (SDN) and edge computing make it possible to send all the IoT generated data to the core where it processes in the racks by moving control for the center out to the edge [12].

Therefore, it is need of the hour to apply intelligent data engineering on data flow management of the big data produced in the enterprise level public and private organizations. In this article, we look at the SDN architecture without traffic engineering and our created traffic engineering mechanism. The comparison is also performed on huge amount of data in University's network. We investigate the SDN architecture while altering, controlling, securing, and maintaining network communication. Furthermore, how to make intelligent decision on flow management while dealing with high volume diversified data in the network of a University. University data is selected due to its intensive usage and diversified nature of data including text, images, video and voice.

Here, Section II presents a brief literate review in the field of software defined networking. Section III describes the experimental setup for our work. Section IV presents detail of proposed methodology. Section V elaborates the results and discussion and in section VI we give brief conclusion regarding the effectiveness of the proposed methodology.

II. RELATED WORK

Traffic management in traditional networks is crucial for the network operator in configuring network. To implement policies for handling networking functions and restricting the operator to display traffic policies is difficult as a high-level intent. SDN provides a new approach for traffic management [13]. This approach is achievable with Procera, an event-driven framework that has four control fields namely; traffic flow, data usage, traffic status and time, and that makes traffic management feasible [14]. In SDN, traffic is managed by introducing novel policy sets for controlling traffic and reducing complexities.

OpenFlow protocol is used for managing traffic flow in SDN by using forwarding devices that are configured on the control plane [9]. In OpenFlow, the forwarding tables of switches have flow table entries installed by the controller, and each entry has a match field, counter, and forwarding actions. SDN uses two approaches, namely, proactive and reactive decisions, to make decisions for flow management. In proactive decisions, the controller installs flow entries in the flow table permanently and before this decision is required [15]. One drawback of this approach is the extra flow table entries that are unnecessary for ternary content-addressable memory. In reactive decisions, the packet arrives at the switch with no forwarding rules, the switch notifies the controller for that packet. The controller identifies the packet path and installs the respective roles in all switches with an identified path after this packet is forwarded to the destination. Reactive decisions work on timeouts. The controller has an expiry time of one second. Reactive flow management reduces large flow tables but increases the reliability of the control channel, control plane [16].

Throughput and latency are the key elements for monitoring SDN performance. An improved throughput is achieved using an aggregate of multiple traffic flows in SDN [17], [18]. Throughput in SDN is measured with a standard tool, *Iperf*. It used TCP/IP flows with TCP windows, and latency is checked using the synthetic frame with various frame sizes and time stamps. The latency in SDN is due to two reasons, processing delay in the control plane and transmission delay of control messages between controllers and switches [19], [20]. In SDN, latency reduces network performance, and overall, 91% of latency is due to the calls to controller APIs instead of transmission of large control messages from the controller to the switches. Latency is the key element in network speed. Network bandwidth encounters a problem under the impact of latency. SDN performs undesirably if the delay source is highly powered and creates a temporary or persistent latency in SDN [21].

III. EXPERIMENTAL SETUP

The experimental setup is based on a comprehensive topology for network traffic, which enables us to optimize the flow control of the packets across the networks. Three diverse cases are based on the number of maximum hosts in the network that produces big data to achieve the noteworthy changes in the figures of following criteria.

1. Latency
2. Throughput

A. SCENARIO 1–UNIVERSITY OF SARGODHA (UOS), QUAID-I-AZAM CAMPUS

This network connects more than 3,000 PCs in topology using 10 controllers and 90 switches with a default bandwidth of 1,000 Mbps between the links within the campus network and the Pakistan Education and Research Network (PERN) connection to link to the Internet. The heavy network traffic was analyzed after capturing the data using renowned sniffing software, wireshark [20]. Data is processed to take decision on flow management, that is, the shortest path.

B. CASE 1(A)

Initially, the captured data is classified into different classes. Thereafter, the data of all these classes is sent to the local area network through the proposed SDN approach, which engineered traffic, for the Quaid-i-Azam campus to measure the latency of each class specified. Furthermore, all the links within the network increased at time T1, and some of the links abruptly collapsed at time T2. Information is updated while monitoring the network. The broken links are re-established, so that a number of information gets updated during the monitoring of the network. This information is added by python script to modify the shortest path between the links.

C. CASE 1(B)

In this case, the data of all the classes are sent outside of the campus network using PERN connection. PERN connection is facilitated with a maximum bandwidth of 36 Mbps. The data of all the classes are sent through the proposed SDN for the Quaid-i-Azam campus to measure the latency time for each reckoned class.

D. CASE 2

For another case, the classified data is used for traffic engineering to record the maximum bandwidth utilization, which is a high throughput gained by the network elements. The data of all classes is sent to the network (node–node) with a maximum bandwidth of 1 Gbps to record the throughput achieved by each class within the proposed SDN. After this experiment, the throughput achieved by the hosts after shaping the traffic is procured.

E. SCENARIO 2–UNIVERSITY COLLEGE OF AGRICULTURE (UCA)

For the network of UCA, which is a constituent college of UOS, more than 1,200 PCs are connected in topology using 7 controllers and 50 switches with a default bandwidth of 1000 Mbps between the links within the campus network and the PERN link to get connected to the Internet. The heavy network traffic is analyzed after capturing the data.

F. SCENARIO 3–SARGODHA MEDICAL COLLEGE (SMC)

Sargodha Medical College (SMC), a constituent of UOS, has a separate campus with a well-equipped computer network for medical students. This institution has more than 500 PCs in topology using 3 controllers and 25 switches with a default bandwidth of 1000 Mbps between the links within the campus network and the PERN association for Internet connectivity.

IV. PROPOSED METHODOLOGY

The experimental setup produced big data which is captured for analysis and processing. As discussed in Section III, scenarios are presented where the heavy network traffic was generated within an enterprise network having more

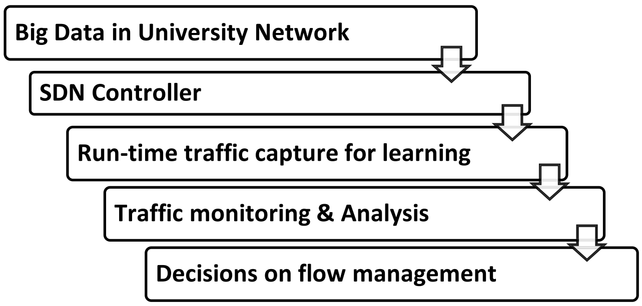

FIGURE 2. Block Diagram of Novel approach to manage the flow.

than 2,000 online hosts requesting for different servers. All the networks in the scenarios produced big data, which are versatile in composition but only the following types of traffic was considered for analysis:

a) Text
b) Image
c) Audio
d) Video

The pseudocode described in Table 1, consists of two major portions. First portion tells mainly about the working of controller which creates the network id, port id and data id for each communication along with record of mac addresses.

For all the requests, it is used to define the paths towards receiver i.e. adding routes. These routes will be saved to be used in next portion as well. In short, this is designed to monitor and analyze the traffic in the OpenFlow switches. The other portion describes the flow management decisions that are taken on the basis of information extracted from earlier step. For each type of traffic paths are being determined dynamically to manipulate the OpenFlow entries in the tables.

A module dumper is used here to dump the flow entries with mac addresses, IP addresses and port id's. This module is invoked after learning stage to take decisions on flow management on the basis of shortest path to achieve the less latency time as well as the best path selection to maximize the throughput between the links of OpenFlow switches. Figure 2 shows the block diagram of the used approach.

V. RESULTS AND DISCUSSIONS

For the experimental setup, we have generated the traffic across all the networks in previous section. There are various types of traffic that would be sent or received. In our experiments, we have sent only four types of traffic shown in table 2.

On the basis of traffic type sent over the networks table 3 presents the packet sizes of each type of traffic which have been sent.

The following results are obtained based on various traffic types and their packet sizes.

Traffic containing image data, which is more than 50% of the overall traffic (sent and received) by the hosts, the network traffic is engineered to provide an optimal path having the least latency. Traffic data of one week from the UOS is used initially to engineer traffic.

TABLE 1. Pseudocode for flow management.

```
Input:     link, MessageLog, SwitchEvent
Output:  BestPath with least latency

Log: = [ ]
MessageLog: = [ ]
i: =0, u <- Reply
if (length(u->link)>0) then
   Switchlog: = []
end if
if (length(u->switch)>0) then
    Log[i]<-(u->switch)
    while (SwitchActive) do
        Request<-(u<-SwitchEvent)
        SwitchLog[i]<-(u<-SwitchRequestSync)
            if (length(u->message)>0) then
                logASyncRequest: = [ ]
                time1<-CurrentTime ( )
                busy: =time1/2
                sendRequest(Destination)
                count: =0
            end if
        end while
end if
while(count<busy) do
    time2<-CurrentTime ( )
        if(time2<time1+count) then
            count+=1
            log[i]<-count
            u<-Reply
            log[i]-<(u->SynchRequest)
            messageLog[i]<-(u->message)
            MinRequestTime:=0
                I: =0
                S: =0
        end if
    end while
 while (switchLog[i]) do
      T=switchLog[i]->requestTime
      if(t<minRequestTime) then
          MinRequestTime=t
          S=i
          I++
    end if
    end while
  BestPath=SwitchLog[s]
```

Experiments showed that the data containing images utilized less latency time after shaping the traffic. Figure 3, presents the experiment results on the transfer latency of

TABLE 2. Types of captured traffic.

Type of traffic	No. of packets (one week)	Percentage
Text	1860000	47.40%
Image	2000000	50.97%
Audio	18000	0.46%
Video	46000	1.17%

TABLE 3. Packet Size of all data types over the network.

Type of traffic	Packet Size without Traffic Engineering [bytes]	Packet Size with Traffic Engineering [bytes]
Text	228	625
Image	185	468
Audio	127	354
Video	85	156

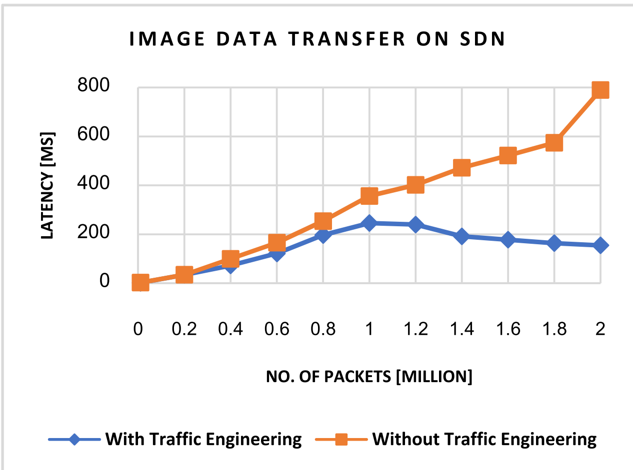

FIGURE 3. Latency of transfer of image data in SDN after shaping the traffic.

image data in SDN. In this experiment, 2 million packets of images are sent to determine the network latency. The blue dots indicate the range from zero to the maximum limit of packets, the data without traffic engineering gradually increased in latency to reach the destination with an increase in the number of packets used. The green dots denote the controlled traffic with engineering. In this stage, 0–1 million packets are used as a learning phase for the system, for which graph shows that the data reached the destination with almost the same latency as those without traffic engineering. However, after 1 million packets a high increase is experienced towards the maximum number of packets, the latency rate gradually declined to a stable position. The experiments validated that big data produced by the network in the form of images with traffic engineering exhibited a low latency rate.

Figure 4 describes the results of the transfer latency of data containing video over SDN with traffic engineering. In video data, a maximum 46880 packets are sent to determine network latency.

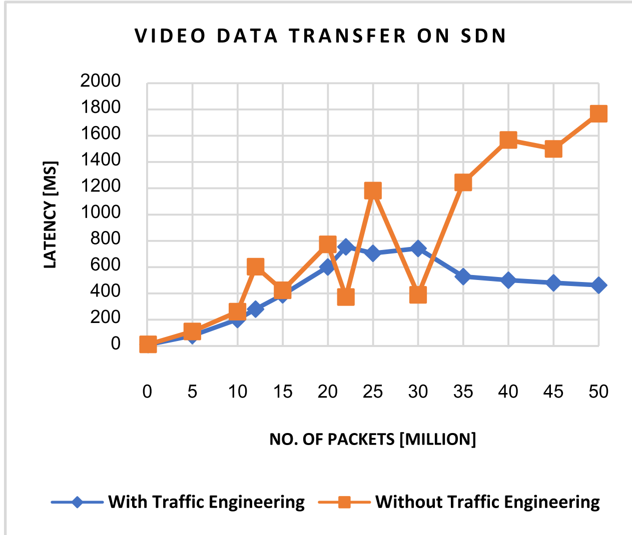

FIGURE 4. Transfer latency of video data in SDN after shaping the traffic.

The blue dots indicate the range from 0 to the maximum limit of packets, the data without traffic engineering showed a zigzag path with a gradual increase in latency as the packets increased. The rise and fall of latency is due to the buffering of video streams. The fall of the blue dots showed the best performance in buffering, and the sudden rise indicates that some packets were waiting with a maximum time resulting in jitters. The green dots on the other hand denote the controlled traffic with engineering. In this stage, half of the data packets, i.e. the initial 22 thousand packets are sent under the learning phase over the network, in which the graph showed that the data reached the destination with a gradual and smooth increase in latency. Thereafter, video streaming packets are stabilized as the latency rate gradually declined. The experiment validated that big data produced by the network in the form of videos with traffic engineering had an excellent latency rate with the proposed approach.

The captured traffic of one month contained a huge amount of text data. Thus, the text data being considered is a major part of the big data produced by the network. Figure 5 presents the results on the transfer latency of text data in SDN after shaping the traffic. In this experiment, a maximum of 1.86 million packets of text data were observed and taken into account to determine the network latency. For the whole-time span, data without traffic engineering showed a gradual increase in latency with some variations at the end which is due to interactive traffic.

As per the controlled traffic with engineering around 0.94 million packets were considered under the learning phase primarily, in which the data reached the destination with a gradual to smooth increase in latency. Subsequently, the latency rate gradually declined for text data with some stability at some points. The experiment validated that the big data produced by the network in the form of text with traffic engineering has an excellent latency rate.

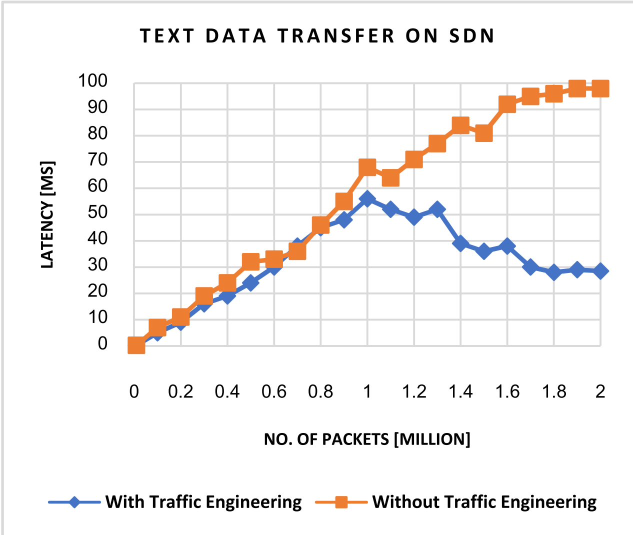

FIGURE 5. Transfer latency of text data in SDN after shaping the traffic.

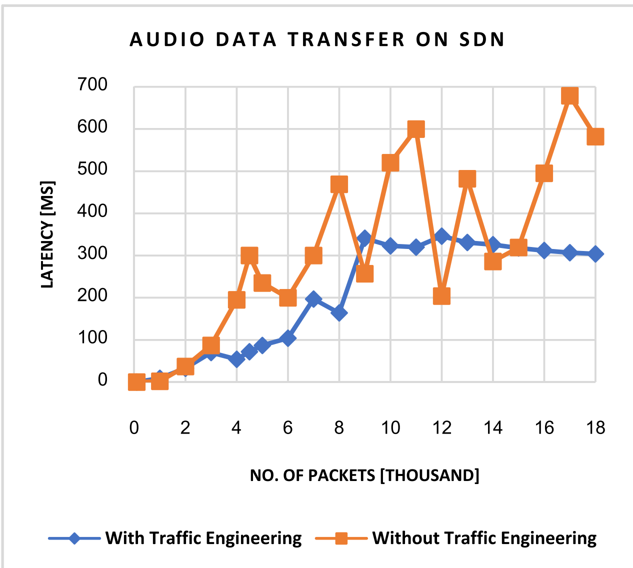

FIGURE 6. Transfer latency of audio data in SDN after shaping the traffic.

Figure 6 presents the results on the transfer latency of audio data in SDN after shaping the traffic. In this experiment, a maximum of 18,000 packets of audio data are sent to determine the network latency. As earlier, the data without traffic engineering showed a zigzag path with a gradual increase in latency till the end. The rise and fall of latency is due to the buffering of audio streams. Largely, without traffic engineering it seems that buffering is being adjusted on demand basis. In the case of traffic engineering with proposed approach for audio data, similar trend is observed in the learning phase, especially the increase from 6,000–8,000 packets, but that should be considered as part of the learning curve. Consequently, the latter half shows smooth transfer of audio data over the network in terms

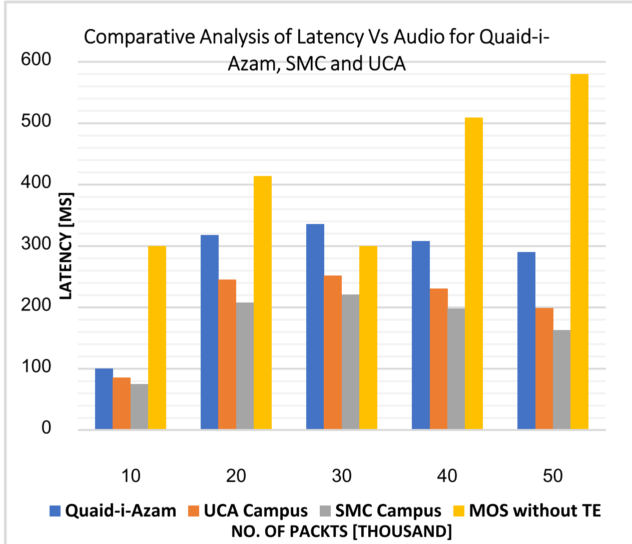

FIGURE 7. Comparative analysis of latency versus audio for the Quaid-i-Azam Campus, SMC and UCA.

of latency. The experiment validated that the big data produced by the network in the form of audio with traffic engineering had a lesser and smoother latency rate than the data without traffic engineering.

A comparative analysis is conducted to measure the latency of audio data among the three campuses of the university. In Figure 7, four different colors indicate the three campuses network load and the mean opinion score (MOS) without traffic engineering. Fifty thousand packets are sent to measure the latency of the data. For every 10,000 packets, latency is recorded to analyze the performance of the networks in different time frames with dissimilar capacities. Based on five recordings, MOS has the highest latency, whereas SMC has the lowest latency, thus indicating that SMC has the best performance. UCA and Quaid-i-Azam Campus ranked second and third, respectively. All campuses have a low latency rate when traffic engineering is applied compared with MOS that is applied without traffic engineering. The variation in campus latency is due to the network size. Quaid-i-Azam Campus has the largest network, followed by UCA and then SMC. All networks are slightly the same with equal latency based on their performance. The result also shows that the network in all campuses is in the learning phase when 30,000 packets are applied, but thereafter latency decreased and became stable.

Similarly, comparative analysis to measure the latency of image data among the three campuses of the university is shown in Figure 8. Four different colors indicate the network load of three campuses and the MOS without traffic engineering. A maximum of 1.9 million packets are sent to measure the latency of the data. For every 400,000 packets, latency is recorded to analyze the performance of the networks in different time frames having diverse capacities. According to the five observations, MOS had the highest latency and SMC had the lowest whereas UCA and Quaid-i-Azam Campus

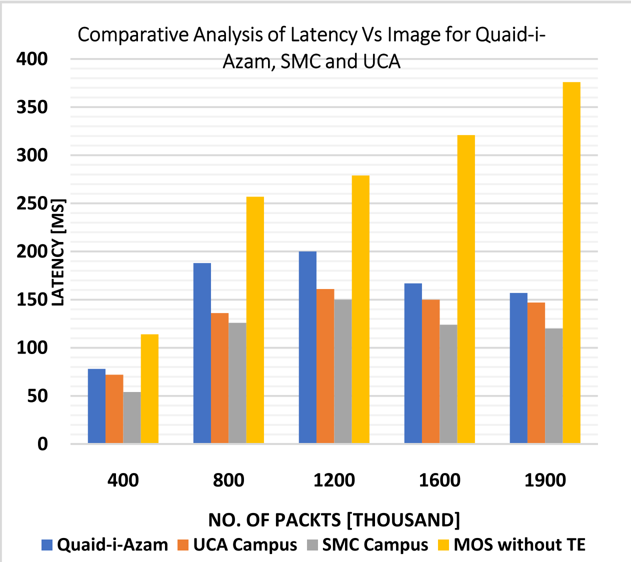

FIGURE 8. Comparative analysis of latency vs. image for Quaid-i-Azam Campus, SMC and UCA.

ranked second and third, respectively. All campuses have low latency rate when traffic engineering is applied as compared to MOS which is running without traffic engineering. The variation in campus latency is due to the network size. Quaid-i-Azam Campus had the largest network, followed by UCA and SMC. All networks were slightly the same with equal latency based on their performance. This graph also shows that the network in all campuses is in the learning phase till 120,000 packets were reached, but the latency decreased and became stable afterwards.

Text data is considered in the set of experiments and compared the results of each site. The latency of text data mainly received by communication of wide area network traffic from three campuses of the university is shown in Figure 9. One million packets are sent to measure the latency of the data. As per the five recordings, MOS has the highest latency and SMC had the lowest latency while UCA and Quaid-i-Azam Campus ranked second and third, respectively. All campuses have a low latency rate when traffic engineering is applied compared with MOS that is applied without traffic engineering. The variation in campus latency is due to network size. Quaid-i-Azam Campus has the largest network, followed by UCA and SMC. This graph also shows that the network in all campuses is in the learning phase when 0.5 million packets were applied, but the latency decreased and became stable afterwards.

In terms of intra-site comparison to measure the latency of video data Figure 10 highlights that a maximum of 7 million packets were sent. For every 1 million packets, latency is recorded to analyze the performance of networks in different time frames with different capacities. Hence, the seven recordings show that MOS has the highest latency and SMC has the lowest latency though UCA while Quaid-i-Azam Campus is quite close in keeping low latency.

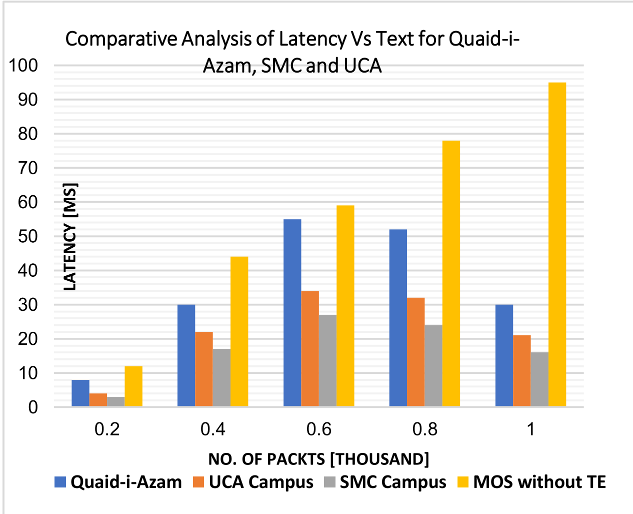

FIGURE 9. Comparative analysis of latency vs. text for Quaid-i-Azam Campus, SMC and UCA.

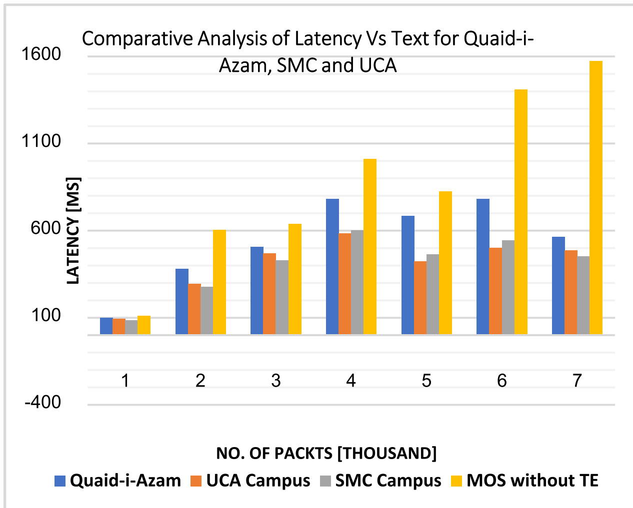

FIGURE 10. Comparative Analysis of latency vs. video for Quaid-i-Azam Campus, SMC and UCA.

This graph also shows that the network in all campuses is in the learning phase till 4 million packets were reached, but the latency decreased and became stable afterwards.

SDN architecture is proposed within the campus with a link capacity among different switches of 1 Gbps as the maximum bandwidth. As shown in Figure 11, after the proposed SDN, the data sent without traffic engineering gradually increase the throughput and obtained a stable position at 300–400 Mbps after 60 seconds. Additionally, four types of data, namely, video, audio, image, and text, are sent to check the throughput with proposed traffic engineering approach. The results show that video data gained the maximum throughput at 500–600 Mbps at a stable position, with the variation caused by the buffering of data. The other factor for the maximum throughput of video data is the larger packet size. Text data is stabilized at 100 Mbps after

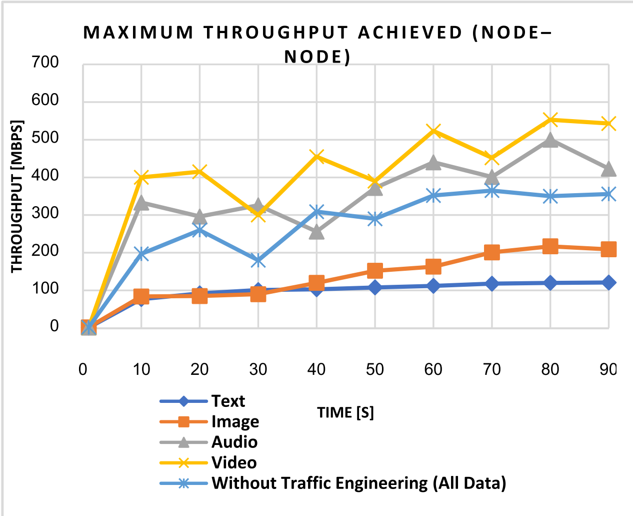

FIGURE 11. Maximum throughput achieved (node–node).

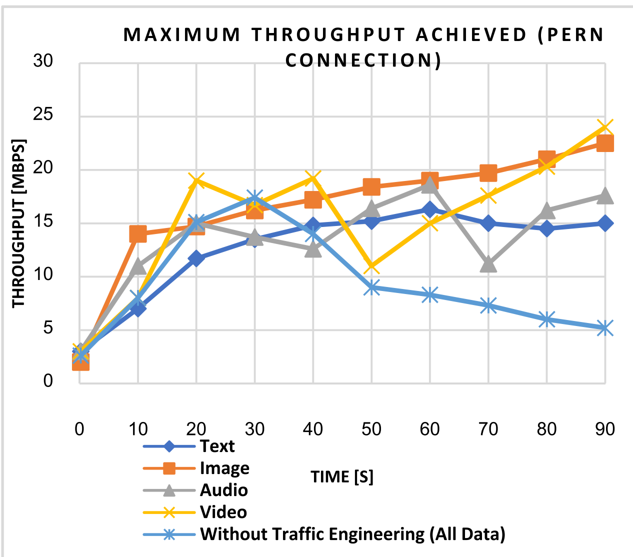

FIGURE 12. Maximum throughput achieved (PERN connection).

20 seconds onward. Image data gradually increased and achieved the maximum throughput of 200 Mbps. Audio data initially achieved a 400 Mbps throughput and showed a stable position at 400–500 Mbps after variations in throughput.

SDN architecture is proposed within the campus using PERN connection provided by HEC for UOS to connect with the global village, that is, the Internet.

PERN connection has a maximum bandwidth of 36 Mbps with dedicated speeds. As shown in Figure 12, after the proposed SDN, the data sent without traffic engineering gradually decreased throughput after 30 s. Additionally, four types of data, namely, video, audio, image, and text, were also sent to check the throughput with traffic engineering applied. The results showed that video data gained the maximum throughput with variation caused by the buffering of data, as discussed earlier. Text data stabilized at 15 Mbps after 40 seconds onward. Image data gradually increased the throughput after 10 seconds while audio data achieved the maximum throughput and showed a stable position at 15–20 Mbps after variations in throughput

VI. CONCLUSION

Initially this research work starts with the generation of the large amount of network traffic as big data by the network. Big data is captured via a sniffer to classify different types of traffic. We gathered the four different types of network traffic on large scale with millions of packets in each type on the basis of their MIME type. Mininet emulator is used with power of Ryu controller to design the topologies and dump the real-time information or status of all the links, hosts and switches. This information is used to take some intelligent decisions on flow management.

This paper has presented that SDN utilize the big data as network traffic to manage the network in terms of flow management with comparatively less latency and maximum bandwidth utilization by the hosts on the basis of traffic type. Hence, the outcomes are reasonable and promise that with better classification of network traffic, improved flow management is conceivable; there are still many other aspects yet to be analyzed. Much space is still there for improvement as far as topology design or the efficiency of the program is concerned. Certain points of possible enhancements and factors can be studied in future.

The improved classification of network traffic can be done on the basis of multiple variables rather than only MIME type, to take strict and precise decisions. It would also help to take priority based resource allocation to the premier members within the network. Additionally, as mentioned earlier we used Class A IPv4 addresses in our experiments which can be replaced to the IPv6 addresses.

REFERENCES

[1] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[2] S. Hassan, J. Sanger, and G. Pernul, "SoDA: Dynamic visual analytics of big social data," in *Proc. Int. Conf. Big Data Smart Comput. (BIGCOMP)*, Jan. 2014, pp. 183–188.

[3] (2017). *The OpenFlow Switch*. [Online]. Available: http://www.openflowswitch.org

[4] D. Kreutz *et al.*, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[5] A. Gelberger, N. Yemini, and R. Giladi, "Performance analysis of software-defined networking (SDN)," in *Proc. IEEE 21st Int. Symp. Modelling Anal. Simulation Comput. Telecommun. Syst.*, Aug. 2013, pp. 389–393.

[6] E. Ahmed *et al.*, "The role of big data analytics in Internet of Things," *Comput. Netw.*, vol. 129, pp. 459–471, Dec. 2017.

[7] W. Tang and Q. Liao, "An SDN-based approach for load balance in heterogeneous radio access networks," in *Proc. IEEE Symp. Comput. Appl. Commun.*, Jul. 2014, pp. 105–108.

[8] S. Yu, M. Liu, W. Dou, X. Liu, and S. Zhou, "Networking for big data: A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 531–549, 1st Quart., 2016.

[9] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[10] H. Silva and A. Neto, "A holistic SDN-capable session-plane tailored for efficient IoMT smart surveillance applications," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2016, pp. 1–6.
[11] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 27–51, 1st Quart., 2015.
[12] A. C. Risdianto and E. Mulyana, "Implementation and analysis of control and forwarding plane for SDN," in *Proc. 7th Int. Conf. Telecommun. Syst. Services Appl. (TSSA)*, Oct. 2012, pp. 227–231.
[13] Y. Guo, Z. Wang, X. Yin, X. Shi, and J. Wu, "Traffic engineering in hybrid SDN networks with multiple traffic matrices," *Comput. Netw.*, vol. 126, pp. 187–199, Oct. 2017.
[14] (2017). *Network Solutions Provider | Procera Networks PROCERA Networks*. [Online]. Available: https://www.proceranetworks.com/2017
[15] H. Kim and N. Feamster, "Improving network management with software defined networking," *Commun. Mag.*, vol. 51, no. 2, pp. 114–119, Feb. 2013.
[16] S. Agarwal, M. Kodialam, and T. V. Lakshman, "Traffic engineering in software defined networks," in *Proc. IEEE Infocom*, Apr. 2013, pp. 2211–2219.
[17] Z. Cai, A. L. Cox, and T. S. E. Ng, "Maestro: Balancing fairness, latency and throughput in the OpenFlow control plane," Dept. Comput. Sci., Rice Univ. Tech., Houston, TX, USA, Tech. Rep. TR-11-07, 2011.
[18] (2017). *Ryu*. [Online]. Available: http://osrg.github.io/ryu/
[19] (Jan. 2018). *Mininet*. [Online]. Available: http://www.mininet.org
[20] Z. Arslan, A. Alemdaroglu, and B. Canberk, "A traffic-aware controller design for next generation software defined networks," in *Proc. 1st Int. Black Sea Conf. Commun. Netw.*, Jul. 2013, pp. 167–171.
[21] M. Caria, A. Jukan, and M. Hoffmann, "A performance study of network migration to SDN-enabled traffic engineering," in *Proc. IEEE Global Commun. Conf.*, Dec. 2013, pp. 1391–1396.
[22] (2017). *Open Networking Foundation*. [Online]. Available: https://www.openenetworking.org

SHAFAAN KHALIQ BHATTI received the M.S. degree in computer science from the University of Sargodha, Pakistan, in 2016. He is currently a Lecturer with the University of Sargodha, Mianwali Sub Campus. His area of research includes computer networks, software-defined networking, cluster computing, and Internet of Things.

M. IKRAM ULLAH LALI received the master's degree in software engineering and the Ph.D. degree in computer science from COMSATS, Islamabad, Pakistan. He was a Visiting Research Fellow with the University of Groningen, Netherlands, under the RuG Fellowship. He was with the Ministry of IT, the University of Education, and the University of Sargodha, Pakistan, for several years. He is currently an Associate Professor with the Department of Software Engineering, University of Gujrat, Pakistan. He is collaborating with different research groups at national and international levels. He has authored over 50 research articles which have been published in conferences and reputed journals. His areas of interests include machine learning, social network data analysis, formal methods, and process models in software engineering. He is a reviewer for many high-impact factor journals and reputed conferences.

BASIT SHAHZAD received the M.Sc. degree from the National University of Science and Technology, Islamabad, Pakistan, and the Ph.D. degree from University Technology Petronas, Malaysia. He was with King Saud University, Riyadh. He was an Assistant Professor with the COMSATS Institute of Information Technology, Islamabad. He is currently a Visiting Scientist with University of Cambridge, U.K., a Visiting Fellow with Macquarie University, Australia, and an Assistant Professor with the Faculty of Engineering and Computer Science, NUML, Islamabad. He is a Collaborating Researcher with the Hagenberg Centre for Software Competence, Austria. His research and teaching career span over 16 years. He has authored or co-authored numerous publications in journals and conferences of international repute and has a very active research profile. He values 'research' a lot and his career objectives revolve around conducting and supervising research in information systems (enterprise architecture, software cost and risk modeling, and mitigating technological risks in modern banking), advancements in research methodologies (quantitative, qualitative, and mixed method), mobile healthcare, and social networks (theory, empirical social network analysis, and microblogging-based analysis). He has an editorial role in several conferences and journals of high repute and has edited numbers of special issues in significant journals in the areas of software engineering, social networks, and mobile healthcare. He is a reviewer for several high-impact journals. He is in the program committee of several distinguished conferences.

FAROOQ JAVID received the M.S. degree in computer science from the University of Engineering and Technology, Lahore, Pakistan. He has been a Lecturer with the University of Sargodha, Pakistan, since 2009. He has authored several research publications in reputed research journals. His area of research includes computer networks, software-defined networking, cluster computing, and Internet of Things.

FAKHAR ULLAH MANGLA received the M.S. degree in computer science from International Islamic University, Islamabad, Pakistan. He is currently an Assistant Professor with the University of Sargodha, Pakistan. He has authored several research articles published in reputed peer-reviewed journals. His area of research includes algorithms, theory of automata, and artificial intelligence.

MUHAMMAD RAMZAN is currently pursuing the Ph.D. degree with the University of Management and Technology, Lahore, Pakistan. He is currently a Lecturer with the University of Sargodha, Pakistan. He has authored several research articles published in reputed peer-reviewed journals. His area of research includes algorithms, machine learning, and computer vision.