ISBN: 978-628-95207-8-1. ISSN: 2414-6390. Digital Object Identifier: 10.18687/LACCEI2024.1.1.1466

Implementation of SDWAN technology over legacy networks for the optimization of packet traffic applying Control of delay and traffic analyzer (CDTA)

Diego Antonio Castro Canales
Faculty of Engineering
UNITEC
Tegucigalpa, Honduras
diegocastro02@unitec.edu

Rolando Arturo Silva Quiñonez
Faculty of Engineering
UNITEC
Tegucigalpa, Honduras
Rolando.silva@unitec.edu.hn

***Abstract*— In current networks, data service provider companies require better speed in transporting packets due to the massive traffic that is generated day after day. Previously, traditional networks required using a certain portion of the sent packet to carry out its transmission. Being that the function of this was to be able to optimize performance in terms of transmission and reception of packets. But it also requires space in the packet to be transmitted, with SD WAN technology by the monitoring and control of packet that are send. Theres no need to have that portion of the packet label to control the transmission that is controlled by software and creates the best route for each packet depending on the priority. The architecture required to create the network will be analyzed with the implementation of SDWAN network equipment for virtualization and analyze their respective transmission speeds, comparing a "legacy" network architecture with the implementation of SDWAN equipment to verify the transmission advantages and Route assignment depending on the packet being sent by the applying the CDTA model to analyze the traffic that is generating, the path it is taking and the delay it is produced to send the message.**

Keywords-SDWAN network; legacy network architecture; packet label; best route; route assignment.

I. Introduction

In this research, the study of the implementation of an SD-WAN network on implemented networks will be carried out considering the network architecture, traffic engineering for the SD-WAN network on legacy networks, comparison of the sending of packets in the old technologies and current management for the SDWAN network for traffic forwarding and control.

For WAN history During Mid to late 1970's X.25 use to be the protocol for WAN. The main characteristic of X.25 was its reliability. X.25 had superior error checking mechanism and allowed virtual circuits. But later it appears another technology at the late 1980s witnessed the advent of ATM by using connection-oriented approach in forming the virtual circuits. After in the 1990s and 2000s appear two technologies that are IPSEC VPN over ethernet as the technology of choice for WAN and Multi-protocol label switching protocol that has an architecture where the labels or Tags are used for switching the data packets. But for the 2010s Software Defined Networks (SDN) technology started raising this concept was to separate the data plane from control plane[1].

The compute power of network switches should be used solely to switch data traffic, while the core intelligence should be stored centrally in a software-based controller. A parallel stream utilizes SDN concepts in a much more basic approach to make the WAN more dependable and user friendly. So, SD WAN concepts are based on SDN, which was used in data center networks to apply a clear separation of control plane and data plane in which forwarding decisions are made, the abstraction of network logic from hardware implementation into software, and the presence of a network controller or Network Operating System (NOS) that coordinates network device forwarding decisions. Because the control and data planes are separated, network switches are reduced to mere forwarding devices, and all intelligence is transferred from routers and switches to software in a logically centralized controller. [2].

SD-WAN is divided into three layers: the data layer, the control layer, and the application layer. This layering strategy separates the wide area network's control plane and data plane, allowing network operators to manage their networks more flexibly and easily. Software-defined wide area networks, as opposed to traditional wide area networks, allow application developers and network operators to articulate their requirements. It turns particular criteria into network configurations that are compliant. [3].

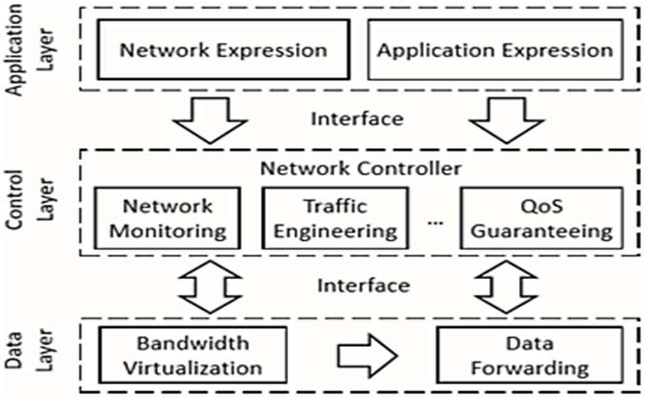


Fig. 1. SDWAN physical architecture

For control plane and data plane we should consider the scalability. At first, we consider for the control plane scalability that has some trouble on this topic. Aggregating and disseminating large quantities of information in real time may easily congest the controller. Second, optimizing the configuration decisions usually involves solving linear problems, which is quite demanding in computation power.

Because it is difficult to treat multiple flows differently by enforcing their own distinct regulations, network operators typically compromise by supporting only a limited number of service classes. Furthermore, the massive network scale exacerbates the dilemma. The number of concurrent flows running through a switch (particularly a core switch) increases as the number of end hosts increases, making Ternary Content Addressable Memory (TCAM) space scarcer. Massive packet losses and stochastic network states will result if suitable methods to assure per-packet consistency during policy updates are not in place.[4].

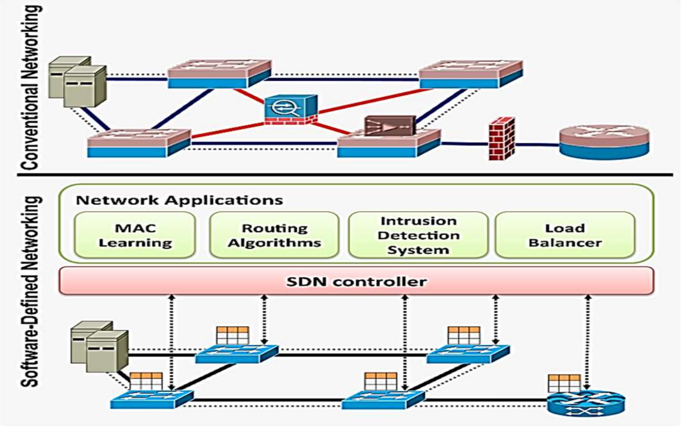


Fig. 2. Traditional Network architecture vs SDN network architecture.

For legacy networks as Fig. 2. Traditional Network architecture vs SDN network architecture. shows like MPLS we know that sending packets to specific network over MPLS, devices check Label Forwarding Information Base (LFIB) tables and examine which label to use for specific network. But for new technologies introducing more services which will call for significant advances and changes in MPLS networks. Networks will have to go in the direction of network programmability, virtualization, and cloud-based services [5].

For legacy network applications that use SDWAN technologies. When SD-WAN is used, the deployed device at each branch is responsible for connecting the branch to any selected access method. We have Internet access methods, private links access methods, or the device may use both. A centralized controller would control the device to determine how traffic will flow. SD-WAN has the potential to significantly reduce MPLS costs. Enterprises can improve automation to eliminate costly routing hardware, reduce network overhead, and improve IT efficiency by reducing the difficulty of managing infrastructure and connectivity and implementing software-based management [6].

Recent research has indicated that by the year 2020, large data or big data, as it is more commonly known, would reach the size of more than 7 zettabyte. One of the biggest hurdles in any DCN is the transmission of big data that has been collected. The reason behind this is that most data centers make use of ECMP (Equal Cost Multipath Protocol) for the transmission of data [7]. The benefits of SDN are realized because of the separation of any network's control and data planes. Traffic engineering is the process of continuously monitoring, dynamically analyzing, calculating, predicting, and specifying data behavior via a network to improve network performance. This methodology allows you to build a network and then apply a set of approaches to improve network performance, management, and security. [8].

The entire performance of a connection, known as Quality of Service (QoS), is an important aspect of networking. QoS includes specifications for all important aspects of data transmission, such as reaction time, jitter, interrupts, and so on. To ensure QoS, application flows must be distinguished because they compete for bandwidth Then network resources must be allocated to ensure the precedence of the higher-priority traffic. That allows for the appropriate network resource distribution [9]. Traffic Engineering (TE) is crucial for network availability and reliability. Enterprises can orchestrate them. Traffic in consideration of the monitoring measurements of WAN performance, such as packet delay, loss, jitter, and service requirements [10]

There are various prerequisites that must be met before SDWAN adoption. Software-defined WANs cannot function adequately by simply expanding a small scale SDN, as they require global knowledge of network traffic. Due to the rich diversity of QoS needs, real-time monitoring, decision making, and policy updating are required, as well as fine-grained control to be capable of regulating traffic at the per-flow or even per-packet level. Make the correct judgment and make the appropriate change to update network policies in a consistent manner, ensuring the atomicity of changes[4].

In Section I we will see the architecture of the SDWAN network and how it works to make work the three layers. The Section II will be explained the way how traffic engineering is done in control plane by the traditional networks and SD-WAN networks and finally in the Section III there is a simulation for an SD controller with different parameters for transmission of each path based on the delay of each traffic to choose the better path in real time and do a real time transmission comparison with legacy networks when there is not separate the control plane for the data plane.

II. SD-WAN NETWORK DESIGN FOR TRAFFIC AND DELAY CONTROL

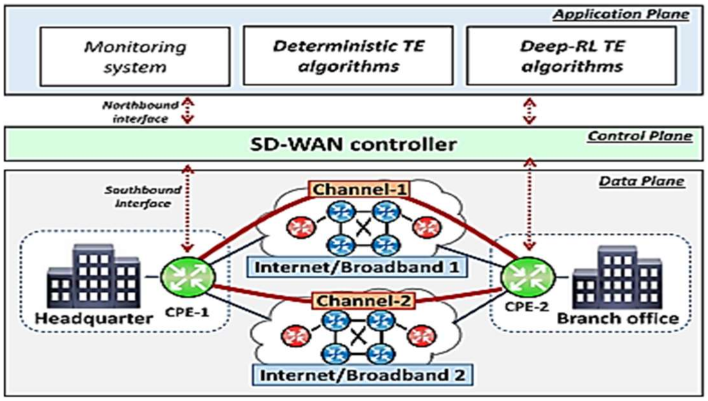


Fig. 3. SD-WAN network architecture. The CPE devices, which are located oversee the routing traffic data into different WAN's.

As shown in Fig. 3, it was possible to observe how the control device used could handle network equipment to assign the ideal route for data transmission through any means. This controller would assign the paths to use based on various parameters set by the user, allowing for monitoring and the application of quality-of-service control.

We will be studying the scenario where a simulation will take place, with two CPE devices implemented to facilitate data transmission. These devices will be connected to a controller that will monitor the network's status and based on the resulting parameters, decide using the algorithm it possesses for network selection.

Each CPE device will introduce a certain level of delay in packet transmission sent to the controller, to determine whether to check the current network status and, depending on that, choose to send the data through the first path or the second path.

A. Network structure

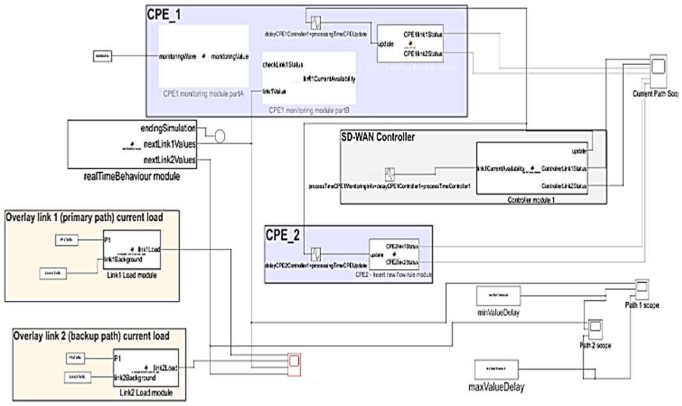


Fig. 4. simulation of SDWAN network with two CPE and one SD-controller

In our network, the load to be sent through the CPE (Customer Premises Equipment) devices responsible for traffic delivery is located. As mentioned, the separation of the control plane and the data plane is established so that a single centralized device will handle the path where each data will be sent, and the data plane will be represented by the CPEs receiving messages from users generated by overlay links 1 and 2. It contains the current network load that will be sent along the path, depending on the parameters imposed by the controller. Next, we have CPE blocks 1 and CPE 2; they will receive updates on the network's status from the controller, which will assign them the path they will take for data transmission.

Then, the SD-controller block maintains the global view of present use of each path in the network using various network measurement technologies, and we can design a traffic scheduling algorithm to dynamically plan data forwarding paths to meet users' requirements[11].

Within the connections being established, they will be verified through different blocks that will graph the state of the network, the amount of traffic being transported on each path, the delay that is occurring, and the status of each stage of packet transmission to determine the current location of the message.

B. Data plane

The two blocks responsible for generating the load sent through the CPEs, this part of the network, are separate from the intermediary devices of the network because they simulate an overlay network, like the tunnel-interfaces, to send traffic through virtual links. Next is the network behavior block, responsible for sending values on how the link is and its availability, depending on the link1Value sent to a CPE, with the latter being the intermediary devices for traffic transmission.

a) Traffic load message

These blocks are composed of 3 components each, with the first one being responsible for generating the new data to be sent. In this case, both have this shared component because we want to transfer the new data generated, regardless of the link to be used. The second component is the accumulated traffic for each link. These two components will be controlled by a module consisting of an algorithm that, depending on the link's status and the status it has in the CPE devices, will assign the output value for each load. If there is no availability, then the value is 0, representing the impossibility of data transmission.

$$L_i = M_i + \sum_{j=1}^{m} Pij * L_j \quad (1)$$

In equation$L_i = M_i + \sum_{j=1}^{m} Pij * L_j$ (1) L_i=The mean arrival rate at each node i in network topology. M_i = The external arrival rate of traffic at each node i in network P_{ij}= Routing matrix.

The equation$L_i = M\Box i\Box + \sum_{j=1}^{m} Pij * L_j$ (1) helps to calculate the traffic sent in the equation where T_{th} represents the actual throughput achieved in the transmission of data, P_p represents a primary path and P_a represents an alternative path.

$$\boldsymbol{T_{th} = (L_i + P_p) + (Li + P_a)} \quad (2)$$

b) Realtime behavior of network

Associating queues with a port allows you to divide the available bandwidth into multiple queues and specify the minimum and maximum speeds for each queue, but this does not guarantee that the available bandwidth will be used efficiently. In addition, special tuning may be required for the behavior of the classifier during overload; otherwise, the minimum throughput may not be met[12].

Equation (3) represents the selection of the primary path and alternate paths. According to the algorithm, if P1 is available before P2, P1 would be chosen as the primary path; otherwise, the path will be selected as the alternate path. Where f_h (P1, P2) is a function representing the paths originating from host and P1is the first channel and P2 is the second channel that will be selected to be choose as a primary and alternative path.

$$\boldsymbol{f_h\,(\mathrm{P1}, \mathrm{P2}) = \begin{cases} P_p \; if \; P1 \; before \; P2 \\ \quad P_a \; otherwise \end{cases}} \quad (4)$$

c) CPE

These blocks are responsible for acting as intermediaries for data transmission. They receive the status of each route from the controller, and depending on the network's status, they allow transmission on the corresponding link. They also have a process to simulate the delay that occurs when updating the information sent by the SD-controller. The difference is that CPE 1 will have two monitoring modules, one for the waveform, which is part A, and one for route availability, which is part B. However, the information on the transmission status of the links will be updated simultaneously.

Software Defined Networking (SDN) technologies removes. software from all networking devices in a network, embeds it into one or more external servers and provides standardized vendor-agnostic interfaces between them[16].

Delay manifests itself in a number of ways, including the time taken to establish a particular service from the initial user request and the time to receive specific information once the service is established[13].

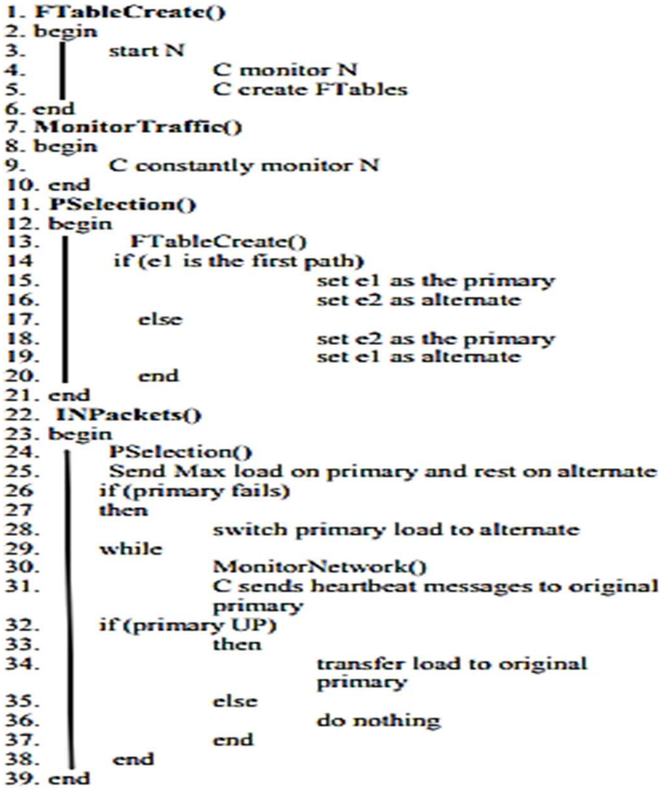
Delay can be manifested by the time taken to establish the service and the equation $Delay_{1-way} = T_2 - T_1$ (2) where T1 represent the time that was sent for the origin and T2 represent the time that was receive.

$$Delay_{1-way} = T_2 - T_1 \quad (5)$$

C. *Control plane*

a) SD-controller

This block is responsible for network management, and, in this case, it has an algorithm to redirect traffic from one path to another. It evaluates various parameters around the network, including network behavior, the maximum and minimum delay values, the data transmission limit, and the network routing procedure[14]. It starts by assessing whether the parameters fall within the assigned values. If they do not, it initially checks the behavior of the first link, and if the network is unavailable, it bypasses link 1 and only sends data through the second link. However, if there is an issue and the network is available, it checks the Delay ranges and data transmission limit. If these exceed the specified ranges, it sends an update message to the CPE and subsequently changes the values to send traffic through path 2. The importance of including delay performance in the control plane lies in the fact that unexpected bursts or collapsed VM-based network functions may request the control plane to recreate a service chaining process for deciding a new service traversal[15]. The Fig. 5 shows the algorithm used in SD-controller.



```
FTableCreate()
begin
    start N
            C monitor N
            C create FTables
end
MonitorTraffic()
begin
    C constantly monitor N
end
PSelection()
begin
    FTableCreate()
    if (e1 is the first path)
                    set e1 as the primary
                    set e2 as alternate
    else
                    set e2 as the primary
                    set e1 as alternate
    end
end
INPackets()
begin
    PSelection()
    Send Max load on primary and rest on alternate
    if (primary fails)
    then
            switch primary load to alternate
    while
            MonitorNetwork()
            C sends heartbeat messages to original
            primary
    if (primary UP)
            then
                    transfer load to original
                    primary
            else
                    do nothing
            end
    end
end
```

Fig. 5 algorithm used in SD-controller.

D. *Application plane*

This SD controller will be sending to the application plane that will be simulated as the scopes that are shown so that is the application of the CDTA model, that shows data collected thorough the traffic to control the packets that are sent and works as an application that shows graphics and works as a monitor for the user to see the performance in both paths.

III. Advantages of SD-WAN Applications Compared to Traditional Networks

For our research study, we will employ a Simulink model that apply CDTA that incorporates various algorithms for traffic generation and its subsequent behavior, including delays that will be calculated by the equation $Delay$⍰$1 -way = T\mathbf{2} - T_{\mathbf{1}}$ (5) that will control the path where the traffic will be send and the establishment of routes through a SD-controller. The routes employed in this context are simulated as an overlaid network, where, despite having two physical routes, they exist logically within a tunnel where signals are transmitted in a manner designed to remain undetectable, much like the way GRE tunnels or VPNs create virtual links.

The generation of traffic initiates from a data matrix that progressively produces data over time, each packet has the maximum of 1600 bytes, where 1500 are used by the MTU parameters and the other 100 bytes are used to simulate the control that the SD-controller will send to update the status of the paths. These data sets accumulate with the data that is meant to be sent via the route, contingent upon the route's deemed efficiency, this is show in $\boldsymbol{T_{th}} = (\boldsymbol{L_i} + \boldsymbol{P_p}) + (\boldsymbol{Li} + \boldsymbol{P_a})$ (2)

Factors such as availability, delay, bandwidth capacity, and queue parameters are verified to facilitate network redirection.

When the network's status is updated by the equation $\boldsymbol{f_h}$ $(\mathbf{P1},\mathbf{P2}) = \ \boldsymbol{P_p}\ \boldsymbol{if}\ \boldsymbol{P1}\ \boldsymbol{before}\ \boldsymbol{P2}$⍰$\boldsymbol{P_a}\ \boldsymbol{otherwise}$ (4) that will send the CPE the primary and alternative path by selecting the first or the second path by the condition that is stablish in the coding that if the delay is higher than 150ms and the link is available, there will be a change of path to optimize and bring full connectivity with the best conditions.

To represent the network's functioning, two traffic transmission routes are established, simulating a connection achieved through two CPEs responsible for directing traffic based on the SD-controller's algorithm. Notably, these CPE devices also possess routing algorithms that function as intermediate routers, carrying out instructions from the control plane, exclusively focusing on the task of traffic transmission. This separation of the control plane from the data plane aims to enhance data handling and efficiency in data transmission, as the presence of a control plane in each device would introduce greater delays. In our model, the SD-controller represents the segment of the control plane under examination. In conventional networks, such separation is absent, and each CPE device would bear the responsibility of managing both the control and data planes, resulting in reduced efficiency and a lack of network centralization.

The core objective of this research is to emulate the operation of an SD-WAN network by employing two CPEs to portray the network's behavior in two distinct scenarios: one featuring the presence of an SD-Controller and the other in its absence. Additionally, the research endeavors to assess the network's performance in both scenarios when both the control plane and data plane reside within routers.

In the first scenario, where the SD-Controller is active, effectively serving as the centralized control plane, it undertakes the management and orchestration of routing policies, selection of optimal paths, and traffic optimization. The two CPEs operate as endpoints within the network and are subject to the SD-Controller's control. The controller makes intelligent routing decisions based on predefined policies and real-time network conditions, facilitating efficient traffic management and dynamic adaptation to network requirements.

In the alternative scenario without an operational SD-Controller, the responsibility for routing and traffic control is delegated directly to the routers or CPEs. Routers or CPEs are tasked with managing routing policies and independently making routing decisions. This approach may lead to a less automated process and increased reliance on manual configuration. The network's performance in this scenario can exhibit greater variability, relying heavily on the skills and configurations of network administrators to optimize traffic. In the absence of the centralized intelligence provided by the SD-Controller, the network may struggle to adapt effectively to the ever-changing network conditions and traffic requirements.

In the alternate scenario where the control plane is managed by the CPEs, each CPE will incorporate a controller to oversee network traffic, thereby intensifying the workload for each device. This approach may occasionally introduce higher delays, potentially impacting network efficiency negatively.

IV. RESULTS

In both scenarios that will be simulated in a computer that have the following specifications:

- Processor: Intel Core i5-4590 3.3Ghz
- Memory ram:12 GB
- Graphic card: Intel(R) HD graphics
- Type of disk: HDD

In the first scenario, the implementation involves having a Software-Defined (SD) controller within our network. In this case, the SD controller, equipped with an algorithm for route re-routing, makes decisions based on the values present in the network. It determines, depending on these values, which link to use for sending packets. In this scenario, the second route is configured to send more data than the first route.

In the first scenario, as depicted in Fig. 6, using an SD controller, one can observe how the data traffic between each route changes. In the model, more data is sent through the second route than the first, as evident in Fig. 7, where the delay of the first route is shown in the yellow-colored graph, and Fig. 8, where the delay of the second route is represented in the green-colored graph. In both figures, it is apparent that the delay between each route is different, and the second route shows a tendency to have lower delay. Where both have the same minimum delay value represented by the blue line and the maximum delay represented by the peach line.

Fig. 9 provides a diagnostic data sent and lost, along with the percentage sent for each route. Despite some losses, the transmission on the second path only experienced a loss of 6.3%, and approximately 95.6% of the data was transmitted through the second route, which offers a better quality of service. This process is carried out using the algorithm implemented in the SD controller to determine the assigned route.

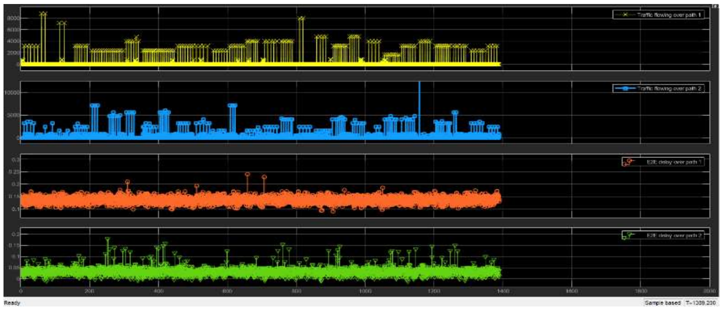


Fig. 6 bytes transmitted in first scenario with SD controller.

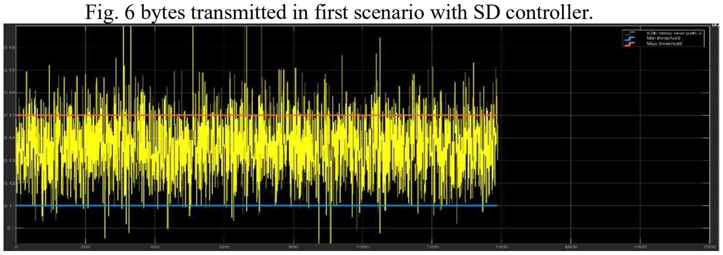


Fig. 7 delay over path 1 in first scenario.

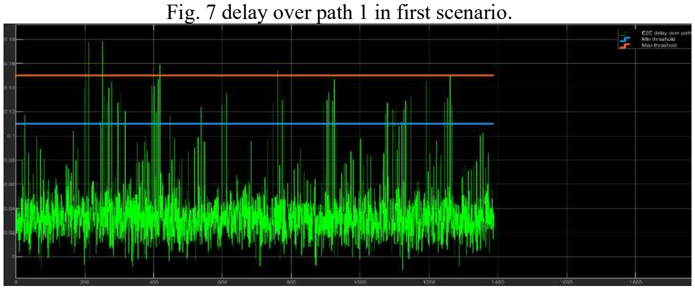
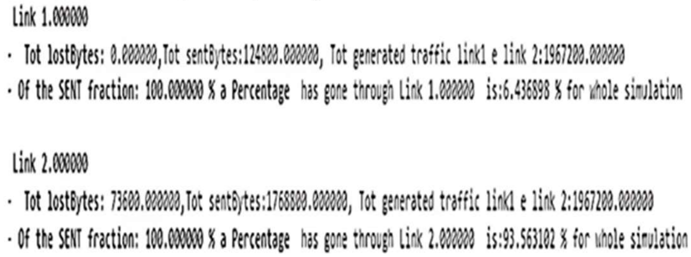


Fig. 8 delay over path 2 in first scenario.



```
Link 1.000000

·  Tot lostBytes: 0.000000,Tot sentBytes:124800.000000, Tot generated traffic link1 e link 2:1967200.000000
· Of the SENT fraction: 100.000000 % a Percentage  has gone through Link 1.000000  is:6.436898 % for whole simulation

Link 2.000000

·  Tot lostBytes: 73600.000000,Tot sentBytes:1768800.000000, Tot generated traffic link1 e link 2:1967200.000000
· Of the SENT fraction: 100.000000 % a Percentage  has gone through Link 2.000000  is:93.563102 % for whole simulation
```

Fig. 9 diagnostic of traffic in first scenario

In the second scenario, where no SD-controller is present, traffic is exclusively routed through a single path, namely the first route. This is evident in Fig. 10 where data is solely transmitted through the first route. There is a higher volume of data transmission compared to the first scenario, as observed in the figure. However, for each transmitted packet, the size remains within a moderate range of 4000 to 6000 bytes. The increased number of transmitted packets is a consequence of all traffic being directed through this single route, leading to the appearance of more closely packed lines in the graph, contrasting with the first scenario.

Nevertheless, this approach adversely impacts the process, as illustrated in Fig. 11, where the delay incurred by having only one route for signal transmission is significantly higher than that experienced in the first scenario. In the first scenario, the minimum delay value represented by the blue line and the maximum delay represented by the peach line. the average delay was approximately 0.15 seconds, while in this case, it extends to approximately 0.30 seconds. This increased delay is attributed to the accumulation of all network traffic on a single link that lacks the sufficient capacity to transmit the data. Fig. 12 provides a diagnostic overview of the data sent through the first link and the associated losses. Although losses were lower by having only a 1.35% of losses that are by 5% lower than in the first scenario, it is crucial to consider the SD-WAN's capability to transmit information to the cloud. Unlike live signal transmission, this information can be retransmitted using various error-checking protocols.

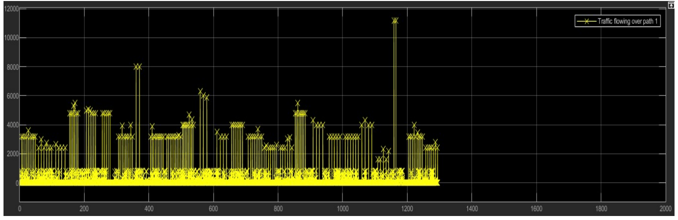


Fig. 10 traffic over path 1 in second scenario

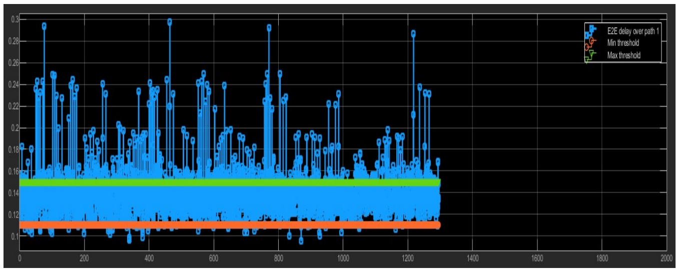
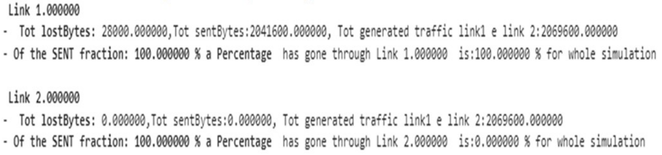


Fig. 11 delay over path 1 in second scenario



```
Link 1.000000
-  Tot lostBytes: 28000.000000,Tot sentBytes:2041600.000000, Tot generated traffic link1 e link 2:2069600.000000
- Of the SENT fraction: 100.000000 % a Percentage  has gone through Link 1.000000  is:100.000000 % for whole simulation

Link 2.000000
-  Tot lostBytes: 0.000000,Tot sentBytes:0.000000, Tot generated traffic link1 e link 2:2069600.000000
- Of the SENT fraction: 100.000000 % a Percentage  has gone through Link 2.000000  is:0.000000 % for whole simulation
```

Fig. 12 diagnostic for second scenario

V. Conclusion

Based on the analysis conducted in both scenarios, we can conclude that, despite experiencing slightly higher losses in the first scenario compared to the second, these losses were minimal due to the processing capabilities of the SD-controller. However, it was evident that the delay in the second scenario was significantly greater, with peaks reaching double the maximum allowable delay. In contrast, the first scenario exhibited low delay, particularly through the second route, where most of the traffic flowed. The SD-controller, equipped with an algorithm, effectively mitigated delays in signal transmission by dynamically changing routes to prevent accumulation and latency.

Although the first path in the first scenario did not incur losses, its substantially lower capacity posed challenges when transmitting large quantities of real-time data. This limitation resulted in significant losses, potentially causing considerable delays and network congestion due to slow data processing.

In a real-world scenario, the presence of an SD controller and routes with enhanced data transmission capacity, particularly for sending data to a server such as a backup in the cloud, proves advantageous. For instance, one could configure one or more routes for lower-capacity traffic, such as communication between end-users or the transmission of lighter data packets, while dedicating another route to handle higher-capacity traffic. Through intelligent algorithms, it becomes possible to identify specific paths.

VI. References

Periodical references

[1] Mohit Chandra Saxena y Preeti Bajaj, «Evolution of Wide Area network from Circuit Switched to Digital Software defined Network», *IEEE*, p. 7, 2021.

[2] Marcin Markowski y Przemyslaw Ryba, «oftware Defined Networking research laboratory – experimental topologies and scenarios», 2016, doi: 10.1109.

[3] Zhenjie Yang, Yong Cui, Baochun Li, Yadong Liu, y Yi Xu, «Software-Defined Wide Area Network (SD-WAN): Architecture, Advances and Opportunities», *IEEE*, p. 9, 2019.

[4] Shuhao Liu y Baochun Li, «On Scaling Software-Defined Networking in Wide-Area Networks», *IEEE*, vol. 20, n.º 3, jun. 2015.

[5] Irena Šeremet y Samir Čaušević, «Advancing IP/MPLS with Software Defined Network in Wide Area Network», 2019.

[6] Khirota Gorgees Yalda, Diyar Jamal Hamad, y Nicolae Ţăpuş, «A survey on Software-defined Wide Area Network (SD-WAN) architectures», *IEEE*, p. 5, 2022.

[7] Muhammad Junaid Anjum, Imran Raza, y S. A. Hussain, «Real-Time Multipath Transmission Protocol (RMTP): A Software Defined Networks (SDN) based Routing Protocol for Data Centric Networks», *IEEE*, p. 6, jul. 2019.

[8] Irena Šeremet y Samir Čaušević, «Advancing Multiprotocol Label Switching Traffic Engineering with Segment Routing in Software Defined Network environment», p. 6, mar. 2020.

[9] R.Gokilabharathi y Dr.P.Deepalakshmi, «Efficient Load Balancing to Enhance the Quality of Service (QOS) in Software Defined Networking (SDN», *IEEE*, p. 6, 2018.

[10] Sebastian Troia, «On Deep Reinforcement Learning for Traffic Engineering in SD-WAN», *IEEE*, vol. 39, n.º 7, p. 15, jul. 2021.

[11] ZHAOGANG SHU *et al.*, «Traffic Engineering in Software-Defined Networking: Measurement and Management», *IEEE*, vol. 4, jun. 2016.

[12] Yuri Ushakov, Margarita Ushakova, y Bolodurina Irina, «Researching the Problem of Providing Quality of Service in Software-Defined Networks», *IEEE*, p. 6, oct. 2021.

[13] Rd Yovi Manova, Edi Sukmadirana, y Nurma Siti Nurmanah, «Comparative Analysis of Quality of Service and Performance of MPLS, EoIP and SD-WAN», *IEEE*.

[14] Yuanguo Bi y Chuan Lin, «Intelligent Quality of Service Aware Traffic Forwarding for Software-Defined Networking/Open Shortest Path First Hybrid Industrial Internet», *IEEE*, vol. 16, n.º 2, feb. 2020.

[15] Yu-Jia Chen y Li-Chun Wang, «Deterministic Quality of Service Guarantee for Dynamic Service Chaining in Software Defined Networking», *IEEE*, vol. 14, n.º 4, p. 12, dic. 2017.

[16] Sachin Sharma *et al.*, «Implementing Quality of Service for the Software Defined Networking Enabled Future Internet», *IEEE*, 2014, doi: 10.1109/EWSDN.2014.36.