Tabriz Journal of Electrical Engineering (TJEE), vol. 52, no. 4, Winter 2022 Serial no. 102
DOI: 10.22034/tjee.2023.15801

ESV-DBRA: An enhanced method for proportional distribution of the multitenant SDN traffic load

Arash Ghorbannia Delavar[1*], Keivan Beigi[2]

[1,2]Department of Computer Engineering and Information Technology, Payame Noor University (PNU), P.O. Box 19395-4697, Tehran, Iran.
E-mail Addresses: a_ghorbannia@pnu.ac.ir[1], k.beigi@zi-tel.com[2]
[*]Corresponding author

Received: 06/20/2022, Revised: 11/04/2022, Accepted: 12/07/2022

Abstract
One of the obvious reasons for most disorders in network service provisioning is network path congestion. Congestion avoidance in today's networks is too costly and sometimes impossible. With the introduction of SDN, centralizing the equipment's control plane has become possible. This paper presents an enhanced method named ESV-DBRA to avoid congestion in multi-tenant SDN networks. At first, ESV-DBRA monitors the traffic load and delay of all network paths for each tenant individually. Then, by merging the parameters obtained from the monitoring, the Service Level Agreements (SLA), and a novel proposed cost function, it calculates the cost of the network paths per tenant. As a result, traffic for each tenant is routed through the path/paths at the lowest possible cost from the tenant's perspective. Next, the bandwidth quotas will be calculated and assigned to the tenants over their optimal routes. Afterward, whenever congestion is likely to occur in a path, ESV-DBRA automatically changes the route or bandwidth of the tenants' traffic related to this path to avoid congestion. Related algorithms are also proposed.
Eventually, simulations show that the proposed method effectively increases bandwidth utilization by 10.76%.

Keywords
Virtual tenant networks (VTN), Software-defined networks (SDN), OpenFlow, Dynamic bandwidth resource allocation (DBRA), Congestion avoidance, Path Cost Estimation.

1. Introduction

Because the bandwidth demand of access network subscribers is increasing due to new and modern centralized applications [1], availability and quality of service (QoS) are becoming more important. Network link congestion is one of the parameters that directly affect these two factors, which may occur for various reasons, such as link failure or improper distribution of network traffic load. Avoiding congestion in traditional networks is costly and sometimes practically impossible. This is due to reserving bandwidth for both the main and backup paths, which is not only expensive but also leads to having so many link capacities idle and useless.

But with the advent of software-defined networks (SDN), in which the control plane of devices is separated from the data plane and centralized control part of the network [29], network management has been simplified, and SDN allows for rapid network innovations [26]. Some researchers believe that SDN has become the de facto standard for future network infrastructures [6] and could execute a re-routing strategy to improve network effectiveness and decrease data congestion [3].

The reason for choosing multi-tenant SDN in the proposed method is that flat SDN platforms often have theoretical and experimental usage. Because in a flat SDN, all servers and clients are placed in the same general environment and are not isolated. It causes many problems in large enterprise networks, such as low security, the multiplicity of broadcast packets in layer two networks, the multiplicity of routing records in layer three networks, and equipment limitations.

Virtual Tenant Networks (VTN) technology is introduced rather than flat SDNs to solve such problems, which allows dividing the physical network into several isolated logical networks. With this approach, the complexity of the underlying network will be hidden, and network resources will be managed more profitably to reduce network configuration time and errors [21].

The multitenancy concept is also used in other environments, such as data centers (DC), service providers (SP), and infrastructure providers, to address resource allocation issues [27]. Due to network resource limitations, the multitenancy concept also plays a significant role in 5G networks. SDN is set to address the limitations of traditional networks [2], and the 5G network slicing architecture is a key technique that enables the operator to create traffic separation that provides optimized solutions for different network users. Network slicing enables the sharing of 5G infrastructure between different tenants, leading to improved service customization and increased operator revenue [4].

DOI: 10.22034/tjee.2023.15801

This slice is introduced as a response to two main challenges: 1) efficient transmission of data flows with radically different characteristics and QoS requirements over the same physical network infrastructure, and 2) providing seamless support for diverse business models and market scenarios, for example, to provide comprehensive communication services to industrial verticals. This slicing architecture is also realized by technologies such as VTN, SDN, Network Function Virtualization (NFV), Multi-Access Edge Computing (MEC), Mobile Edge Computing, Cloud/Fog Computing, and so on [5-9].

The proposed method, Enhanced SDN VTN Dynamic Bandwidth Resource Allocation (ESV-DBRA), offers a proportional distribution of the multi-tenant SDN traffic load. ESV-DBRA monitors the bandwidth utilization of each VTN and compares it with the determined SLA. Then it takes the needed actions automatically when it guesstimates any path (or paths) congestion before occurrence. The actions include finding the next best path per VTN and calculating the bandwidth volume of each VTN that should be moved to the next best path. Each path cost for VTN is calculated from end to end to find the best path for each VTN traffic, and the path with the lowest cost is selected as the best path for that VTN. The proposed method offers a novel bandwidth and path delay function to calculate the path cost.

As industry needs and user experience become more sensitive and rigorous to latency timeliness, the development trend of SDN has shifted the focus from the pursuit of stability and compatibility to the satisfaction of low-latency communications [10]. Accordingly, in addition to the bandwidth parameter, the delay of the network paths should also be calculated accurately so that the route selection process can be more accurate. Therefore, the path delay is calculated end-to-end using an innovative method in the proposed method.

Furthermore, network bandwidth resources are dynamically allocated for each VTN based on the SLA. Finally, after calculating the best path and the new bandwidth for each VTN in each link, the new bandwidth values are checked, and if they match the free capacity of the links concerned, they are applied to the physical links. This method is constantly repeated if necessary to prevent possible bandwidth congestion in the network.

2. Path congestion avoidance using a DBRA system in a VTN-support SDN

The most commonly used VTN-supported SDN controller is OpenDaylight (ODL), which by default uses Dijkstra's algorithm to select the best path. This algorithm, also known as Shortest Path First (SFP), does not consider future changes in bandwidth utilization while choosing the best path. If traffic increases suddenly, network congestion may occur due to the inconsideration of bandwidth overhead and lead to a decrement in the packet transmission rate. In other words, all packets are only forwarded through the path defined at the initial calculation, and so it causes network performance degradation [7].

Fig. 1(a) shows an example of bandwidth congestion in which VTN routing is performed using the Shortest Path First (SPF) algorithm that allocates all VTN traffic to the same path with the minimum hop count. So, in this scenario, path-1 will always be selected for all VTN and, therefore, will be congested after a while. The method presented in this paper proposes an enhanced Dynamic Bandwidth Resource Allocation (DBRA) system to meet such problems in SDNs with multiple logical tenants, as shown in Fig. 2(b).

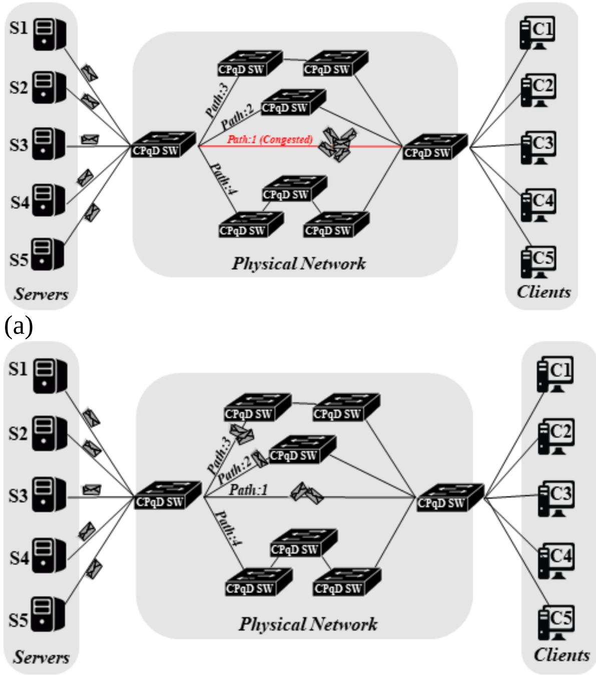


Fig. 1. An example of path congestion avoidance using a DBRA system in a SDN datacenter. (a) Using the default SPF algorithm, (b) Using a DBRA system.

3. Related Works

Some management methods have been proposed and researched to deal with bandwidth congestion and guarantee the quality of service (QoS) based on SDN.

For example, in [4], some methods have been compared regarding deploying distributed and demand-based network functions, service-guaranteeing network cutting, flexible coordination of network functions, and optimal workload allocation. These methods generally consist of SDNs, NFV, and Edge Computing [19].

Some other methods also support multi-tenancy capability, such as [5], which offers a multi-tenant MEC management architecture with support for network and application slicing.

Group-Constrained Shortest Path (GCSP) is another method to dynamically allocate the network resources in the VTN-support SDNs, which proposes a new batch-optimization framework where resource allocations for a small group of flows are performed simultaneously as the number of new service requests and network conditions vary.

But it helps only the parties involved in the delivery of OTT video services, such as the Network Service, Providers (NSP) and Video Service Providers (VSP) [11].

DOI: 10.22034/tjee.2023.15801

Some other SDN methods and techniques are compared in [12], such as AutoSlice, OpenVirteX, FlowN, Network Hypervisor, NVP, AutoVFlow, and libNetVirt. Each method presents a hypervisor that places and works as a proxy between the SDN switches and controllers. Since these hypervisor-based solutions add an extra layer to the system, they have their disadvantages; they lead to complications in the previous system and consume more system resources. However, solutions have been proposed to increase the efficiency of SDN environments, especially for more complex solutions such as hypervisor-based SDNs.

For instance, [30] offers a method called FreqScal to reduce the power consumption of OpenFlow switches based on setting the frequency of the switch according to its traffic load.

Another algorithm, with the help of creating virtual domains, tries to improve the resource utilization rate and reduce intra-domain communication delays compared to the SPF and Greedy Resource Mapping algorithms [13]. It does not consider the concern of multitenancy and imposes a single point of failure on the system. Multiple virtual networks will be affected if a single underlying network device fails.

Another DBRA system named Software-Defined Dynamic Bandwidth Management (SDBM) has been proposed using a weighting policy in a combination of the OpenFlow architecture and the OpenWrt platform. Firstly, it monitors and analyses networks and the statuses of applications and then dynamically re-allocates resources with software-defined discipline [14]. SDBM has only been implemented in flat SDNs and does not consider multitenancy.

Also, some DBRA systems are proposed only for optical networks, such as the SDN-based Optical Virtual Private Network (OVPN) [1], or New Generation Ethernet Passive Optical Network (NG-EPON) [15], and the Extension of Software-defined Optical Network Slicing Architecture (eSONA) [18]. These methods, based on the user's requested SLA, decide whether the system requires additional bandwidth resources or not.

Another DBRA method for a VTN-support SDN is a Routing Planning (RP) mechanism for determining the best path in a physical network and a Bandwidth Resource Planning (BRP) mechanism designed to manage the VTN's needs for bandwidth resources and thereby improve the network bandwidth resource utilization [6]. This system improves the best path selection and bandwidth adjustment based on the network conditions and increases bandwidth utilization per VTN compared to the SFP algorithm. But this method can only be used when the path delay is very low. Because in the proposed cost function in the RP/BRP-DBRA method, the delay and the cost variables are inversely proportional, as the path delay increases, the cost decreases, and vice versa, which is not desirable and loses its efficiency. Also, the path delay is not collected in real-time. On the other hand, only unidirectional bandwidth utilization (traffic sent from the clients to the servers) is considered. This is while congestion may also occur in the server's outbound direction.

The method proposed in this article has been tried to address the issues and weaknesses of [21]. Previously in [20], it has been attempted to drive a relative improvement in the path cost calculation in a multi-tenant SDN and an improvement in upload traffic distribution towards the RP/BRP-DBRA. But in this article, by proposing the ESV-DBRA, an enhanced method is presented for calculating the path costs and obtaining the path delay values in real-time, with a novel efficient method. A summary of the related works is listed in Table I for easy comparison.

4. Proposed Architecture

ESV-DBRA performs calculations to predict and eliminate the possible congestion of the SDN links. This method, currently designed for Fixed-SDNs (Non-Mobile SDNs with fixed nodes), automatically senses the excessive traffic load in any SDN path before occurrence and arranges the necessary measures and calculations to remove the possible congestion.

For networks with mobile nodes, more than one controller is needed to avoid unnecessary complications due to the existing delivery process, processing many messages in the operational environment, the complexity of resource management, load balancing, etc. Fixed SDN has been used in calculations and implementation and clarifying the results. In this way, the traffic load of each of the VTNs is proportionally distributed on the links with optimal service quality and less delay.

Table I. Comparing the Related Works

Proposals	Objective	Use Case	Used Technology	Cons & Limitations
[19]	Traffic Management	5G	SoDeMa	Doesn't support Fixed-SDN
[5]	Application Service Management	5G	MEC	Doesn't support Fixed-SDN
GCSP [11]	Resource Management	Video services	Batch-optimization of DRA	Designed only for video services providers.
[12]	Resource Management	Virtualization	Hypervisors and NFV	Add an extra layer to the system and will cause complications and consume more system resources.
FreqScal [30]	Resource Management	General	FPGA and Frequency scaling	Limited to power resource management.
[13]	Resource Management	Big Data	Virtual Data Domain	Doesn't support VTN and makes a SPOF
SDBM [14]	Resource Management	General	Weighting Policy	Doesn't support VTN
[1]	Resource Management	Optical Networks	NG-EPON	Limited to optical networks
[15]	Resource Management	Optical Networks	OVPN	Limited to optical networks
eSONA [18]	Resource Management	Optical Networks	SONA	Limited to optical networks
RP/BRP DBRA [21]	Resource Management	General	Min-Cost	Limited to the lower delay values (doesn't work properly with higher delay values),

Tabriz Journal of Electrical Engineering (TJEE), vol. 52, no. 4, Winter 2022 Serial no. 102
DOI: 10.22034/tjee.2023.15801

4.1. Proposed Method Innovations

Fig. 2 shows the general structure of the ESV-DBRA system. This system is placed between the network admin and the SDN controller. It means the ESV-DBRA system performs routing calculations using the SLA values and the monitoring results (including bandwidth utilization and delay) for each VTN separately. Based on the routing results, it is possible that all the traffic of a VTN passes through the same path, or be is distributed unevenly among several different paths, depending on the physical network conditions.

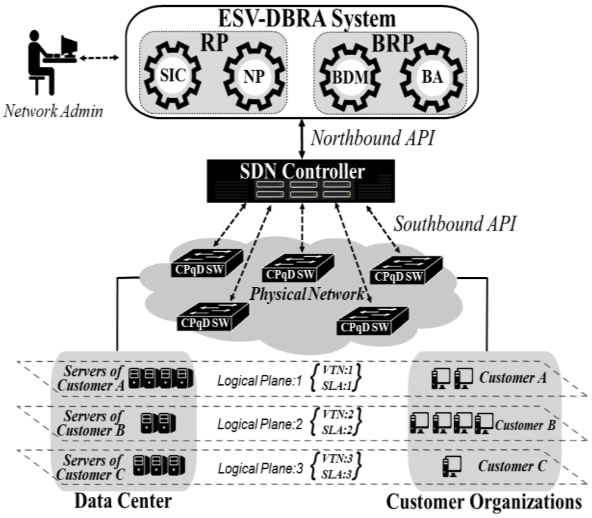


Fig. 2. Overall structure of the proposed ESV-DBRA system on a VTN-support SDN.

Finally, the routing results are sent to the SDN controller using its native Northbound APIs, and the SDN controller instantly applies them to the relevant equipment in the physical network using the related Southbound APIs. The ESV-DBRA system also receives some information (including delays and both inbound and outbound traffic loads over the links) from the controller at a specific time interval to monitor the physical condition of the network and makes subsequent decisions by performing calculations against the gathered information.

Another innovation of the proposed method is calculating the path delay using an inventive probe-based technique.

Due to the existing limitations of the native software switch (Open vSwitch or OVS) in bandwidth provisioning on the SDN software switch interfaces, the CPqD software switches ("Centro de Pesquisa e Desenvolvimento em Telecomunica¸c˜oes" or also known as ofsoftswitch13) have been used, which have specific REST APIs to update the switch interface bandwidth.

Although the CPqD software switch solved the bandwidth-provisioning problem, it also led to other limitations. There is a lack of a proper dedicated REST API for monitoring and receiving statistical data about the link delay and the bandwidth utilization of interfaces in an instant [16]. Therefore, a probe is designed to obtain a relatively high approximate delay of each path per VTN. The higher delay accuracy causes us to calculate the more accurate path cost and then make a better decision. This feature gives the proposed method the ability to cover more SDN sizes, while this capability is often ignored in traffic engineering (TE) and dynamic resource adjustment mechanisms.

For example, the delay value in the computations is avoided or assumed as a constant across all of the links in a path. The ESV-DBRA system includes two main modules, one for routing and another for allocating bandwidth resources. Each module includes two separate but interrelated algorithms described in detail in the following subsections.

4.2. RP Module

The RP Module is generally responsible for receiving and checking the inputs from the network operator and then calculating the best route per flow. This section itself consists of two algorithms illustrated in the following.

4.1.1 SIC Algorithm

Since there is no unique measurement technique available in SDN that can authenticate the provided services' satisfaction level, there is always a gap between the service provider and consumer to have service satisfaction information that can be considered for service provider selection decision-making and generate a lack of transparency among service providers and consumers [22]. For this purpose, in ESV-DBRA, we have tried to consider the basic requirements and the main parameters desired by the network admin as much as possible.

The SIC algorithm gathers and checks information based on the SLA requirements per VTN as listed in Table II, and sends a message to the next algorithm, NP, to adjust the routing mechanism.

Table II. Definition of the SLA variables and their acceptance conditions.

Var.	Description	Condition
α	BW Threshold Factor	$0 < \alpha < 1$
β	BW Provision Factor	$\beta > 1$
γ	BW Real Factor	$0 < \gamma < 1$
K	No. of VTNs	$K \in$ N
P_v	No. of Paths per VTN	$(P_1,P_2,...,P_v) \in$ N
H_v	No. of Servers per VTN	$(H1,H2,...,H_v) \in$ N
Sp_v	No. of Switches per Path per VTN	$S_{11},S_{21},...,S_{p1},S_{12},S_{22},...,S_{pv} \in$ N
RBW_v	Requested BW per VTN (bps)	$(RBW_1,RBW_2,..., RBW_v) > 0$
RD_v	Max Requested Delay per VTN	$(RD_1,RD_2,...,RD_v) > 0$
NBW_p	Nominal BW (bps)	$NBW_1,NBW_2,..., NBW_p > 0$

Tabriz Journal of Electrical Engineering (TJEE), vol. 52, no. 4, Winter 2022 Serial no. 102
DOI: 10.22034/tjee.2023.15801

Dp_v	Delay per Path per VTN	$D_{11},D_{21},...,D_{p1},D_{12}, D_{22},...,D_{pv} > 0$

Table III. Definition of the other used variables

Var.	Description
P	The path *p*
$P_{congest}$	The congested path
BW_p	Actual Bandwidth of the path *p*
$BWremain_p$	Remained actual Bandwidth of the *p*
D_p	Total path delays of the links and switches of the path *p*
Cp_v	The cost of the VTN *v* in the path *p*
BP_v	The better path for the VTN *v*
ABW_v	Assigned Bandwidth to the VTN *v*
ABW_{vp}	Assigned Bandwidth to the VTN *v* in the path *p*
$ABWv_{pcongest}$	Assigned Bandwidth to the VTN *v* in the congested path
$ABWi_{pcongest}$	Assigned Bandwidth to the VTN *i* in the congested path
$ABWv_{pothers}$	Assigned Bandwidth to the VTN *v* in all paths except the congested *P*
ABW_{hv}	Assigned Bandwidth to each server in the VTN *v* in the better path
$TABW_p$	Total Assigned Bandwidths to the VTNs on the path *p*
$TABWp_{congest}$	Total Assigned Bandwidths to the VTNs on the congested path
$ABWp_{congest}$	An Array including all Assigned Bandwidths to the different VTNs on the congested path
$ABW_{otherVTNp}$	Assigned Bandwidth to a VTN on the path *p* (except the VTN which has the max *RBW* on the path *p*)
$TABW_{otherVTNp}$	Total Assigned Bandwidths to all VTNs (except the VTN which has the max *RBW* on the path *p)*
$TRBW_{otherVTNp}$	Total Requested Bandwidths to all VTNs (except the VTN which has the max *RBW* on the path *p)*
NBP_v	The Next Better Path for the VTN *v*
$P_{TABWgeBW}$	The path which its $TABW_p$ is greater than or equal to the BW_p of this path (called the Overloaded Path)
$ABW_{overloaded}$	The max *ABW* of VTNs which are in the Overloaded Path
$VTN_{overloaded}$	VTN which has the *ABW* overloaded
t	Throughput measuring time
T_{old}	The previous Upload/Download Throughput measuring time
T_{new}	The new Upload/Download Throughput measuring time
Tu_p	Upload Throughput of the path *p*
Td_p	Download Throughput of the path *p*
UL_p	Upload bandwidth of the path *p*
ULp_{old}	Upload bandwidth of the p at the T_{old}
DL_p	Download bandwidth of the path *p*
DLp_{old}	Download bandwidth of the *p* at T_{old}
ΔUL_p	Upload Bandwidth difference of the path *p*, between the T_{old} & T_{new}
ΔDL_p	Download Bandwidth difference of the path *p*, between the T_{old} & T_{new}
ΔT	Duration between the T_{old} &T_{new}
Uu_p	Upload Utilization of the path *p*
Ud_p	Download Utilization of the path *p*
$kp_{congest}$	Number of VTNs within the congested path
D_p	Delay of the path *p*
Dp_t	Delay of the path *p* at the time *t*
VTN_p	All VTNs which they pass over *p*

Algorithm 1. SIC (SLA Information Collection)

input : Topology and SLA information data per VTN and per PATH
output : Confirmed topology and SLA info per VTN

```
Getting the GENERAL-VARIABLES: α, β, γ, K, P
for i IN GENERAL-VARIABLES do
    if i meets the SLA conditions then
        accept i
    end
    else
        return False
    end
end
Getting Per-VTN variables: RBW, RD, H
RBW, RD, H ← 0
for v = 1 to K do
    if RBW_v, RD_v, H_v meet the SLA conditions then
        RBW ← inserted RBW_v
        RD ← inserted RD_v
        H ← inserted H_v
    end
    else
        return False
    end
end
Getting Per-PATH variables: S, NBW
S, NBW ← 0
for p = 1 to P do
    if S_p, NBW_p meet the SLA conditions then
        S ← inserted S_p
        NBW ← inserted NBW_p
    end
    else
        return False
    end
end
if no False returned then
    send the gathered data to the Algorithm NP
end
else
    exit
end
```

The implemented scenario involves some clients and servers positioned in five different VTNs, using four

separate paths. The total number of paths for all VTNs is considered equal to be more convenient. The other variables used in this paper are described in Table III. While one of the SLA requirements violates the system constraints, an error message is returned to the operator.
Otherwise, it sends a message containing the approved SLA information and other received data to the NP algorithm in the RP module. The corresponding pseudo-code is shown in Algorithm 1.

4.1.2 NP Algorithm

The purpose of the NP algorithm is to dynamically provide an efficient route for each VTN, according to the SIC algorithm outputs and the real-time traffic load. The following steps are as follows:
The work starts with getting two messages; one from the SIC algorithm about the requirements of the SLAs requested per VTN, including the *GENERAL*, *Per-VTN*, and *Per-PATH* variables; and another message from the BDM Algorithm, including the delay sensed by each VTN. Afterward, the cost of the v^{th} VTN in the p^{th} path will be optimally calculated by the proposed objective function as Eq. (1). In practice, the maximum usable bandwidth of a physical link will always be less than the nominal bandwidth reported by the network equipment manufacturers. So, the BW Real Factor (γ) is used to adjust the maximum usable bandwidth of each path.
Also, in regards to getting the final result of the cost function between 0 and 1, the factor of "0.5" is used. This formula is calculated for each path per VTN, ($p*v$) times. So, the result will be a $p*v$ matrix. Then the best path per VTN is calculated by Eq. (2).

$$C_{pv}=\left(\frac{RBW_v}{\gamma * NBW_p}+\frac{D_p}{RD_v}\right)0.5 \quad (1)$$

$$BPv=\text{path}(\min(C_{pv})) \quad (2)$$

Eq. (3) calculates the bandwidth allocated to the desired VTN in the Best Path (*BP)*, based on the total number of existing servers within the v^{th} VTN.

$$ABW_{hv}=\frac{(RBW_v*\beta)}{H_v} \quad (3)$$

To improve the quality of the bandwidth provisioning, the BW Provision Factor (*β)* is used to allocate bandwidth to a VTN more than the requested in SLA (e.g. if the RBW_v=500Mbps and β=1.5, so the ABW_v=750Mbps). Using Eq. (4) total allocated bandwidth of VTNs on the path p will be estimated.

$$TABW_p=\sum_{v=1}^{k_p} ABW_{v_p} \quad , \quad (<p=BP>) \quad (4)$$

$$TABW_p \leq BW_p \quad , \quad (<p=BP>) \quad (5)$$

$$NBP_v=\text{path}(\min(C_{pv})) \quad , \quad (<p\neq BP>) \quad (6)$$

If condition Eq. (5) does not meet the requirements of the desired VTN in the *BP*, then another *BP* called Next Best Path (*NBP*) for that VTN should be determined. Eq. (6) defines the *NBP* based on the lowest cost after the previous *BP*.
There is no concern about the bandwidth resources of one route being used by another VTN because the ESV-DBRA system itself has done all the bandwidth allocated to all VTNs. If the condition Eq. (5) matches for all VTNs in the *BP* path, a message will be sent to the controller containing the amount of assigned bandwidth to each VTN. Finally, the VTNs use their allocated/re-allocated bandwidth quota.

The relevant pseudo-code is shown in Algorithm 2.

4.3. BRP Module

This module is responsible for detecting and removing possible congestion across the network. To carry out this task, first, it should know the current statistics of the network links. Using this knowledge, it can make the right decisions about problems arising from improper bandwidth distribution. Accordingly, it requires real-time monitoring of the physical network.

Algorithm 2. NP *(Network Policy)*

Input: SLA data from Alg. SIC, *D* from Alg. BDM
Output: Minimum path cost per VTN

Function best_path_calculation()
Getting GENERAL, Per-VTN and Per-PATH variables from Alg. SIC
$BW \leftarrow 0$
for $nbw \in NBW$ **do**
$BW \leftarrow (nbw * \gamma)$
end
$ABW \leftarrow 0$
for $rbw \in RBW$ **do**
$ABW \leftarrow (rbw * \beta)$
end
for $abw, h \in (ABW, H)$ **do**
$ABW_h \leftarrow (\frac{abw}{h})$
end
for $(bw, d) \in (BW, D)$ **do**
$C_v \leftarrow 0$
for $(rbw, rd) \in (RBW, RD)$ **do**
$C_v \leftarrow round\ ([C_{pv}=\left(\frac{RBW_v}{BW_p}+\frac{D_p}{RD_v}\right)0.5], 2)$
end
$C \leftarrow C_v$
end
Function *bp*()
for $v \in$ (1 to *K)* **do**
$BP \leftarrow min(C_v)$
end
end
for p = 1 to P **do**
$TABW \leftarrow$ sum of VTNs which has BP=p
end
while *count* (*OverLoaded_Path*) > 0 **do**
for (*tabw*, *bw)* ∈ (*TABW*, *BW)* **do**
$P_{TABWgeBW}$ ← finding paths which (*tabw*>*bw*)
end
for $p \in P_{TABWgeBW}$ **do**
if $p \notin OverLoaded\_Path$ **then**
$OverLoaded\_Path \leftarrow p$
end
$ABW_{overloaded} \leftarrow$ max *ABW* of VTNs in *p*
$VTN_{overloaded}$ ← find VTN with max *ABW* in *p*
Cost of the $VTN_{overloaded}$ = 999
for $v = VTN_{overloaded}$ **do**
while $(BW_v - TABW_v) < ABW_v$ **do**
Cost of the v = 999
end
end
$BP \leftarrow bp()$
end

DOI: 10.22034/tjee.2023.15801

end
end

Therefore, if it forecasts or detects any link/links congestion, it calculates the most optimum possible situation to address the congestion immediately, using the SLA and monitoring results (network statistics). The BRP module has evaluated the calculation results before being sent to the SDN controller. This module consists of the following two algorithms.

4.2.1 BDM Algorithm

This algorithm performs real-time monitoring of the network and calculates the bandwidth utilization and delay per VTN by the following ordered tasks:

1) Sending requests to the SDN controller using the following Northbound REST APIs to get the links' capacity the upload (servers outbound) and download (servers inbound) throughputs and the delay of the *BP*s.
2) Getting and parsing the REST responses.
3) Calculating the Tu_p and Td_p, the server upload and download throughput, through the path p, at the time t, using Eq. (7) and Eq. (8).

$$T_{up}(t)=\left(\frac{UL_p(t)-UL_{p_{old}}}{t-T_{old}}\right)=\left(\frac{\Delta UL_p}{\Delta T}\right) \quad , \quad <t=T_{new}> \qquad (7)$$

$$T_{dp}(t)=\left(\frac{DL_p(t)-DL_{p_{old}}}{t-T_{old}}\right)=\left(\frac{\Delta DL_p}{\Delta T}\right) \quad , \quad <t=T_{new}> \qquad (8)$$

The ΔUL_p is the duration between the two upload throughputs measured at the times t and T_{Old}, in seconds. Accordingly, the ΔT is the measuring interval in seconds. In the same way, the $DL_p(t)$, DL_{Pold} and ΔDL_p are related to the download throughput.

4) Uu_p and Ud_p computation, which are the Upload and Download bandwidth utilization of the path p at time $t2$, obtained by the Eq. (9) and Eq. (10). To round the results to only two decimals, the "*round()*" function is used.

$$U_{u_p}(t)=\text{round}\left(\left[\frac{T_{u_p}(t)}{BW_p(t)}\right], 2\right) \qquad (9)$$

$$U_{d_p}(t)=\text{round}\left(\left[\frac{T_{d_p}(t)}{BW_p(t)}\right], 2\right) \qquad (10)$$

5) Whenever the transmitted or received bandwidth through a path reaches or exceeds the determined threshold, and if this path is utilized by more than one VTN, the condition Eq. (11) has been met. So the BDM algorithm sends a message to the BA algorithm to inform it about the detected possible congestion. This message includes the parsed data from the REST response, including the delay, DPs, and the total assigned bandwidth to each path. Otherwise, no action will be taken.

$$[(U_{u_p}(t)\geq \alpha) \text{ OR } (U_{d_p}(t)\geq\alpha)] \text{ AND } (k_{p_{congest}}\geq 1) \qquad (11)$$

6) As mentioned in section 4.1, due to the lack of capability to get the exact delay value in the used controller and software switches, the end-to-end delay of each path is calculated on the physical links between clients and servers with a novel probe-based method.

Due to the current natural fluctuation in the delay time, caused by, for example, a momentary traffic burst, especially in large-scale networks such as enterprises or service providers, the Weighted Moving Average (WMA) of the delay is used. In this equation, DP is the WMA of the delay of path P, d_{Pt} is the delay of the path P at time t, and W_t is the weight of the t^{th} delay in the moving window. WMA is used to reduce the effect of momentary bursts of the delay time in the routing cost calculations and, finally, in the best route selection.

In the delay time series, the last delay data takes the higher weight, and the older delay data takes the lower weight. The results are reported to the ESV-DBRA system (to the BDM algorithm in the BRP module) by the mentioned probe at interval of 1 second and a window size of 10 seconds. Eq. (12) shows the WMA of the delay for the path p. Algorithm 3 illustrates the above computation procedures done by the BDM.

$$D_p=\left(\frac{\sum_{t=1}^{10} W_t * d_{p_t}}{\sum_{t=1}^{10} W_t}\right) \qquad (12)$$

Algorithm 3: BDM (Bandwidth and Delay Monitoring)

Input: α from Alg. SIC
Output: Congested Paths
Function congestion_detection()
  DL,UL,T ← 0
  for *p* ∈ *Edge_SW_Internal_Links* **do**
    DL_{Old} ← *DL*
    UL_{Old} ← *UL*
    T_{Old} ← *T*
    // running a REST query against the ODL controller to fetch the statistics of *p*
    if *Rest_Result* was OK **then**
      // pars the *Rest_Result*
      DL ← the parsed received bits count as the download volume
      UL ← the parsed the sent bits count as the upload volume
      T ← the parsed time in second as the time
      $Tu_p(t) \leftarrow \left(\frac{UL_p(t)- UL_{Old_p}(t_{old})}{t - T_{Old}}\right) = \left(\frac{\Delta UL_p}{\Delta T}\right)$
      $Td_p(t) \leftarrow \left(\frac{DL_p(t)- DL_{Old_p}(t_{old})}{t - T_{Old}}\right) = \left(\frac{\Delta DL_p}{\Delta T}\right)$
      $Uu_p(t) \leftarrow round\ \left(\left[\frac{Tu_p\ (t)}{BW_p\ \ (t)}\right].2\right)$
      $Ud_p(t) \leftarrow round\ \left(\left[\frac{Td_p\ (t)}{BW_p\ \ (t)}\right].2\right)$
      if ($Uu_p \geq$ α) or ($Udp \geq \alpha$) **then**
        if count (*VTNs* ∈ *Congested_Paths*) ≥ 1 **then**
          Matched_Congested_Paths ← *p*
        end
      end
      else
        return False
      end
    end
  end
  if len (*Matched_Congested_Paths*) ≥ 0 **then**
    send an event to Alg. BA: "These Paths $(*Matched_Congested_Paths*) are congested and have more than 1 VTN"
  end
  else
    return "There is no congestion point on the network now."
  end
end
Function get_delay()

DOI: 10.22034/tjee.2023.15801

Getting the *DELAY* file via SCP from the Mininet VM to the Local_dir
    for $d \in DELAY$ **do**
        $D_p = \left(\frac{\sum_{t=1}^{10} W_t * d_{pt}}{\sum_{t=1}^{10} W_t}\right)$
    end
end

4.2.2 BA Algorithm

This algorithm recalculates the bandwidth quotas for each VTN whenever it receives a message about a congested path from the BDM algorithm. It ultimately sends a message containing the results to the NP algorithm. The BA algorithm steps are as follows:

1) Receiving the BDM algorithm message about the path congestion and selecting the path with the higher cost value, if more than one path is congested.

2) Then, among the VTNs in the congested path, BA selects the one with the most allocated BW.

3) BA determines whether this VTN has *ABW* in the other paths, indicating whether the other bandwidths are reserved for this VTN. Thus, a separate *ABW* is calculated for each VTN per path. If this VTN has *ABW* on other paths, the *ABW* of the other paths is summed, and the product of the total *RBW* of this VTN by 0.9 is subtracted from the sum, as Eq. (13). Then the remainder is assigned to the VTN as the *ABW* of this VTN in the congested path.

$$\mathrm{ABW}_{\mathrm{v}_{\mathrm{p}_{\mathrm{congest}}}} = (\mathrm{RBW}_{\mathrm{v}} * 0.9\ \beta) - \mathrm{ABW}_{\mathrm{v}_{\mathrm{p}_{\mathrm{others}}}} \quad (13)$$

4) Calculating $TABW_p$, which is the sum of the *ABW*s of the VTNs utilizing the congested path and calculated exactly as before, with a bit difference. The path *p* is the congested path, not the *BP* path.

5) Eq. (14) calculates the sum of the other *ABWs* of the VTNs along the congested path (all VTNs except the one having the most *RBW).* Then the difference between the allocated bandwidth of the VTN over the congested path and the total bandwidth of the *p*, calculated as the remaining bandwidth of the path *p*, as Eq. (15).

$$\mathrm{TABW}_{\mathrm{otherVTN}_{\mathrm{p}}} = \mathrm{TABW}_{\mathrm{p}_{\mathrm{congest}}} - \mathrm{ABW}_{\mathrm{v}_{\mathrm{p}_{\mathrm{congest}}}} \quad , \quad < p = p_{congest} > \quad (14)$$

$$\mathrm{BW}_{\mathrm{remain}_{\mathrm{p}}} = \mathrm{BW}_{\mathrm{p}} - \mathrm{ABW}_{\mathrm{v}_{\mathrm{p}_{\mathrm{congest}}}} \quad , \quad < \mathrm{p} = \mathrm{p}_{\mathrm{congest}} > \quad (15)$$

6) Eq. (16) compares the results of Eq. (14) and Eq. (15) to find whether the remaining bandwidth on the congested path is sufficient for $TABW_{otherVTNp}$ or not. If not, BA divides the remaining bandwidth among the other VTNs on this path relative to their requested bandwidth (RBW). But if it is sufficient, so goes to the next step.

$$\text{if } \mathrm{TABW}_{\mathrm{otherVTN}_{\mathrm{p}}} \leq \mathrm{BW}_{\mathrm{remain}_{\mathrm{p}}} \quad , < \mathrm{p} = \mathrm{p}_{\mathrm{congest}} > \text{ then: go the next step} \quad \text{else } \mathrm{ABW}_{\mathrm{up}} = \left(\frac{\mathrm{TRBW}_{\mathrm{otherVTN}_{\mathrm{p}}}}{\mathrm{TABW}_{\mathrm{otherVTN}_{\mathrm{p}}}}\right) \mathrm{BW}_{\mathrm{remain}_{\mathrm{p}}} \quad (16)$$

7) In the last step, the updated *ABW* info is sent to the NP algorithm as a message, to re-execute the *BP* and other related calculations. Finally, the calculation results are applied to the network. Algorithm 4 shows the above calculation steps.

5. Performance Evaluation

Results have been compared with three previous methods, namely SPF, RP-DBRA, and RP/BRP-DBRA, to evaluate the performance of ESV-DBRA. The SPF method determines routes using the maximum hop count and lacks intelligence or dynamics in network bandwidth allocation. The RP-DBRA method provides an algorithm for routing packets per tenant. The RP/BRP-DBRA method, in addition to an algorithm for routing packets per tenant, also provides dynamic bandwidth allocation per tenant [21].

In comparing the four methods based on utilization, path P_1, which has the lowest cost, is selected first in both download and upload directions. If an algorithm has been used to optimize the load distribution, then gradually increasing the traffic volume, the other paths have been utilized. Now, congestion detection speed has the most important role in bandwidth utilization. The faster detection leads to more bandwidth utilization and fewer disruptions.

Algorithm 4. BA (Bandwidth Assignment)

Input: Congested Paths from Alg. BDM, SLA data from Alg. SIC.
Output: New_ABW_c (New assigned BW for the *Matched_Congested_Paths* per VTN)

Function new_abw_mcp()
// Calculating the new Assigned BW of all of the *Matched_Congested_Paths*.
    if count (*Matched_Congested_Paths*) ≥ 1 **then**
        $P \leftarrow P_{congested}$ with $\max(C_p)$
    end
    else
        $P \leftarrow P_{congested}$
    end
    $v = 0$
    $v \leftarrow$ finding the VTN which has max *ABW* in *p*
    //*Creating a Matrix for ABW from the ABW array.*
    Matrix_ABW ← Calculating the *ABW* per VTN
    if *v* has *ABW* on the other paths **then**
        $ABW_{v_{p_{congest}}}$ = round $((RBW_v * 0.9\ \beta) - (ABW_v - ABW_{vp}).2)$
    end
    else
        $ABW_{v_{p_{congest}}}$ = round $((RBW_v * 0.9\ \beta).2)$
    end
    $TABW_{p_{congest}}, TABW_{otherVTN_p} = 0$
    $TABW_{p_{congest}} \leftarrow$ finding the Total *ABW* on the *p*.
    $TABW_{otherVTN_p} \leftarrow$ finding the Total *ABW* to all VTNs except the targeted VTN which has the max *RBW* on the path *p*.
    $BW_{remain_p} \leftarrow$ finding the Total Remained BW on the *p*, $[BW_p - ABW_{v_{p_{congest}}}]$
    if $TABW_{otherVTN_p} \geq BW_{remain_p}$ **then**
        for $i \in VTN_p$ **do**
        //VTN_p is meaning that all VTNs which they pass over the *p*.
            if $i \neq v$ **then**
                $ABW_{otherVTN_{p_i}}$=round $\left(\frac{RBW_i * BW_{remain_p}}{TABW_{otherVTN_p}}\right)$
                $ABW_{i_{p_{congest}}} \leftarrow ABW_{otherVTN_{p_i}}$
            end
            else

$ABW_{i_{P_{congest}}} \leftarrow ABW_{v_{P_{congest}}}$
end
$ABW_{P_{congest}} \leftarrow ABW_{i_{P_{congest}}}$
end
end
return $ABW_{P_{congest}}$
end

5.1. Evaluation Testbed

This section describes the platform used for comparing and evaluating the different approaches. All four methods mentioned have been implemented on the same simulation platform. To cover the network topology of a small and medium-sized business (SMB), the same topology as shown in Fig. 1 is used as the simulation topology, including five VTNs (K=5) and four physical paths between clients and servers of each VTN (P_v=P=4). But, due to the proposed system scalability's importance in implementing ESV-DBRA, a study has been conducted on different SDN implementations and controllers. Mininet, the most popular simulator for SDN resource allocation, has been selected as the simulator. Also, ODL, the most reliable and scalable SDN controller with the ability to support VTN, has been selected [17, 23].

In addition to Mininet, two other machines have been used, one for the ODL controller and another for the VTN coordinator, as listed in Table IV, which also this architecture addresses the system scalability.

5.2. Evaluation Methodology

First, the download (incoming traffic to the servers) bandwidth utilization per path for different methods has been compared. The path utilization and path delay is measured per VTN for 15 minutes to perform the simulation. Fig. 3 compares the bandwidth utilization per path among the different methods mentioned. Fig. 4 is the same as Fig. 3, but about uploading (outgoing traffic from the servers).

In Fig. 3(a), using the SPF method, all VTNs are assigned to *P1*, and so, *P1* has been congested after about 5s, and while the other paths remain useless. In Fig. 3(b), using RP method, due to the lack of a module to distribute the VTNs' traffic load on the paths dynamically, RP only changes the *BP*. It migrates the total traffic load to the new selected path and practically does not solve the congestion problem.

In Fig. 3(a), all VTNs are assigned to P_1. Therefore, P_1 has been congested using the SPF method after about 5s while the other paths remain useless. In Fig. 3(b), using the RP method, due to the lack of a module to distribute the VTNs' traffic load on the paths dynamically, RP only changes the *BP*. It simply migrates the total traffic load to the newly selected path and practically does not solve the congestion problem.

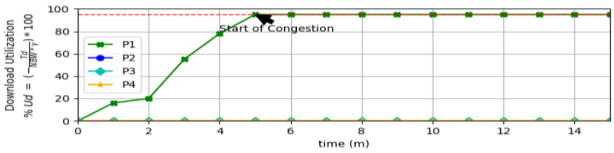
Table IV. Hardware platforms used for simulating the proposed ESV-DBRA method.

VM	OS	RAM	CPU
Mininet	Ubuntu 18.04.2LTS	4GB	4Cores
ODL	Ubuntu 18.04.3LTS	16GB	8Cores
VTN Coordinator	CentOS 7.6.1810	16GB	8Cores

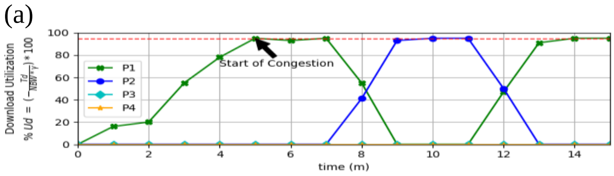
Table V lists the used sample inputs for the SIC algorithm in the proposed RP module (RBW and NBW are in kbps, RD is in milliseconds). Also, the iPerf3 tool has been used to generate bidirectional UDP traffic at the desired speeds.

Table V. Sample inputs of the SIC algorithm for the simulated network topology.

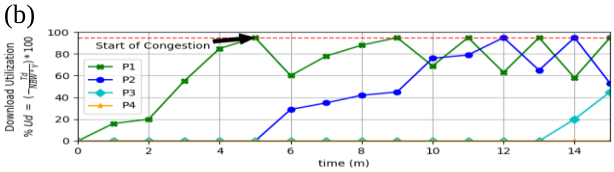
Input Description	Var.	Sample Input
BW Threshold Factor	α	0.85
BW Provision Factor	β	1.1
BW Real Factor	γ	0.95
No. of VTNs	K	5
No. of Paths	P	4
Requested BW of VTN 1	RBW_1	700
Max Requested Delay of VTN1	RD_1	900
No. of the Servers in VTN1	H_1	1
Requested BW of VTN2	RBW_2	500
Max Requested Delay of VTN2	RD_2	400
No. of the Servers in VTN2	H_2	1
Requested BW of VTN 3	RBW_3	500
Max Requested Delay of VTN3	RD_3	1000
No. of the Servers in VTN3	H_3	1
Requested BW of VTN4	RBW_4	300
Max Requested Delay of VTN4	RD_4	700
No. of the Servers in VTN4	H_4	1
Requested BW of VTN5	RBW_5	200
Max Requested Delay of VTN5	RD_5	500
No. of the Servers in VTN 5	H_5	1
No. of the Switches in Path 1	S_1	2
Nominal BW of Path 1 in kbps	NBW_1	1000
No. of the Switches in Path 2	S_2	3
Nominal BW of Path 2 in kbps	NBW_2	1000
No. of the Switches in Path 3	S_3	4
Nominal BW of Path 3 in kbps	NBW_3	1000
No. of the Switches in Path 4	S_4	5
Nominal BW of Path 4 in kbps	NBW_4	1000



(a)

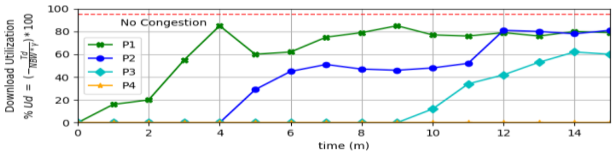


(b)



(c)

DOI: 10.22034/tjee.2023.15801



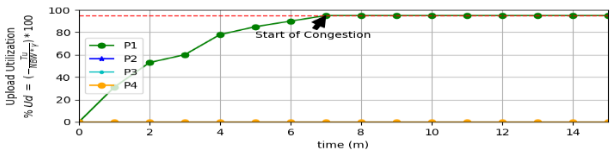
(d)

Fig. 3. Comparison of download bandwidth utilization per path between different methods. (a) SPF, (b) RP, (c) RP/BRP-DBRA, (d) ESV-DBRA.

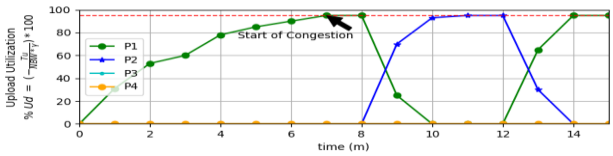
In Fig. 3(c), using RP/BRP-DBRA method, after 5 minutes, congestion and disruption is observed on the P_1 for all VTNs. Then it moved a part of the traffic from the P_1 to the next best path, the P_2. Despite an algorithm to dynamically change the VTN traffic in this scenario, P_1 has been congested 5 times, during the 15 minutes test. Each congestion is followed by a disruption and packet loss in VTN traffic, which is undesirable.

Fig. 3(d) is related to the proposed ESV-DBRA method. After reaching the threshold of the P_1 in the 4th minute, when there is still no disorder in the network, ESV-DBRA immediately transmits a specific volume of traffic from the P_1 to the P_2 and prevents any subsequent disruption. Then, as all VTNs traffic increases, at the 9th minute that the P_1 reaches its threshold, more traffic has been transmitted from the P_1 to another best path other than P_2 (the P_3), which eventually leads to a proper load balance and prevents any following disruption.

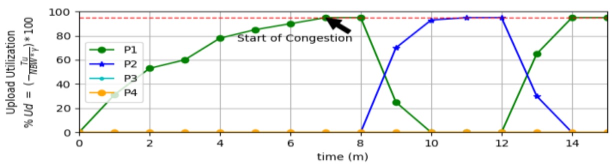
Fig. 4 shows the simulation results based on the upload bandwidth utilization per path for different methods. Figures 4(a) and 4(b) are similar to Figures 3(a) and 3(b) and show the congestion that occurred at the 7th minute.



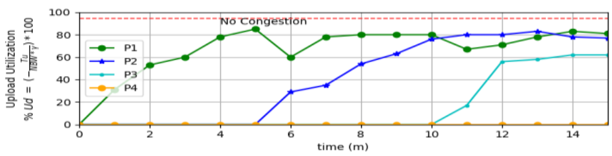
(a)



(b)



(c)

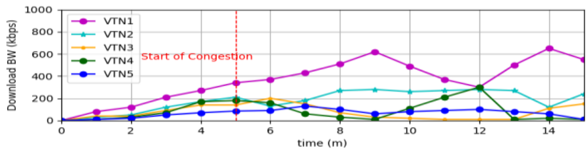


(d)

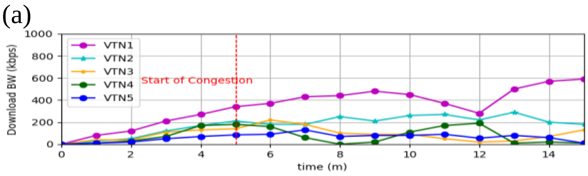
Fig. 4. Comparison of upload bandwidth utilization per path between different methods. (a) SPF, (b) RP, (c) RP/BRP-DBRA, (d) ESV-DBRA.

Congestion occurred in the fifth minute of the RP/BRP-DBRA method, as shown in Fig. 4(c), and traffic migration occurred immediately. But following the congestion, most of the VTNs have been affected, and their traffic has dropped (a little), and it has been resolved after a minute. This happened again in the next four traffic migrations.

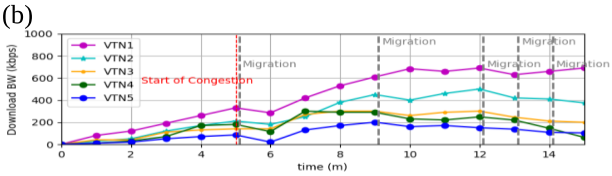
But in Fig. 4(d), which is for the proposed ESV-DBRA system, except for two minor cases, the download bandwidth utilization increases in all VTNs during 15 minutes. Also, in the 4th minute, when the congestion is predicted, traffic migration starts before the occurrence of the congestion. As a result, no congestion, bandwidth drop, or packet loss occurred in any of the VTNs. Fig. 5 shows the simulation results based on the download bandwidth utilization per VTN for different methods.



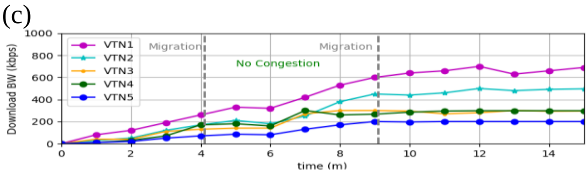
(a)



(b)



(c)



(d)

Fig. 5. Comparison of download bandwidth utilization per VTN between different methods. (a) SPF, (b) RP, (c) RP/BRP-DBRA, (d) ESV-DBRA.

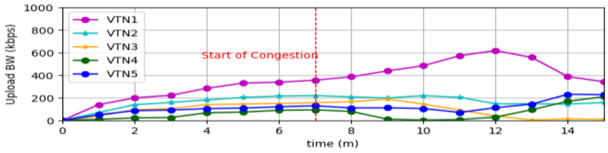
Fig. 5(a), Fig. 5(b), and Fig. 5(c) show some bandwidth drop for the different VTNs at the 5th minute when the P_1 is congested. In Fig. 5(c), even though migration has occurred, some bandwidth drops are observed at any migration (due to the lack of any prediction method). But in Fig. 5(d), which is related to the ESV-DBRA, there is no congestion, so no bandwidth drops at the migration points.

Fig. 6 compares upload bandwidth utilization per VTN among different methods.

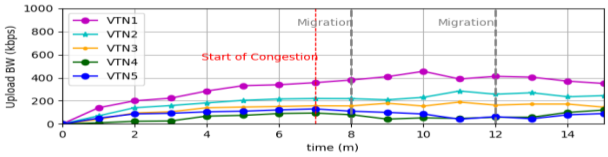
Fig. 6(a), Fig. 6(b), and Fig. 5(c) are just the same as Fig. 5(a), Fig. 5(b), and Fig. 5(c), respectively, but their congestion time is the 7th minute. They show some bandwidth drop for the different VTNs at the 7th minute when the P_1 is congested. In addition, in Fig. 6(d), which depicts the proposed ESV-DBRA system, the upload bandwidth utilization increases continuously in all VTNs,

DOI: 10.22034/tjee.2023.15801

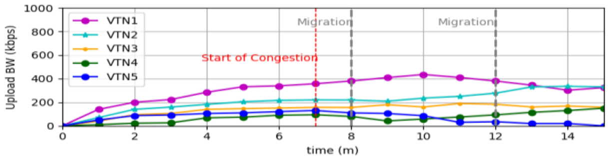
with no congestion, bandwidth drop, or packet loss observed in any of the VTNs. The upload test results are almost the same as the download test results, but with a slight difference. The RP/BRP-DBRA method does not react to the upload traffic and works exactly like the RP method.



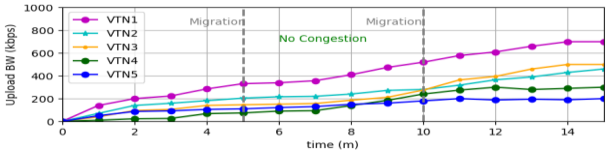
(a)



(b)



(c)



(d)

Fig. 6. Comparison of upload bandwidth utilization per VTN between different methods. (a) SPF, (b) RP, (c) RP/BRP-DBRA, (d) ESV-DBRA.

Likewise, to better understand the effect of ESV-DBRA on network bandwidth utilization, the total inbound traffic volume per VTN has been measured using different methods. According to the results shown in Table VI, using ESV-DBRA, total inbound bandwidth utilization is increased by about 10.76% compared to the RP/BRP-DBRA method, the most successful method among the compared methods.

Table VI. Total inbound traffic volume (KBytes/s) per VTN using different methods

Method VTN	SPF	RP	RP/BRP	ESV-DBRA
VTN-1	41213	38775	46088	49500
VTN-2	20288	23063	27769	33600
VTN-3	8513	9225	22838	28050
VTN-4	10538	9338	19950	18188
VTN-5	7688	7313	13088	14363
TOTAL	88239	87713	129731	143700

6. Future Works

The following studies are considered future works:
- Provisioning a new module to enable the QoS feature for a multitenant SDN and attaching it to the proposed ESV-DBRA system, to perform separate routing and admission control for discrete subsets of traffic such as voice, video, and data;
- Provisioning the static and dynamic VTN Prioritization;
- Provisioning some per-service probes to monitor the different services;
- Using the new parameters such as "reliability" in the route calculation mechanism.

7. Conclusion

ESV-DBRA is a novel method proposed to dynamically allocate bandwidth to the network nodes/links in a VTN-supported Fixed-SDN environment. This method includes two modules and four algorithms to perform calculations to predict and eliminate possible SDN link congestions using the proposed objective function. Also, it takes into overloading when recalculating and adjusting resources per VTN.

The bandwidth utilization has been measured to evaluate the performance of ESV-DBRA while using different methods. The achieved measurement results regarding the use of ESV-DBRA in comparison with three previous methods: SPF, RP-DBRA, and RP/BRP-DBRA show that the proposed system and algorithms effectively increase the bandwidth usage by clients on different tenants. On the other hand, no congestion or bandwidth drop occurred in any VTN traffic during the 15-minutes simulation while using the ESV-DBRA. According to the simulation results, using ESV-DBRA, total inbound bandwidth utilization is increased by about 10.76% compared to the RP/BRP-DBRA method, the most successful among the compared methods.

8. References

[1] A. Rafiq, M.F. Hayat, M.U. Younus, "Dynamic Bandwidth allocation algorithm for avoiding Frame rearrangement in NG-EPON", Optical Switching and Networking, vol. 43, pp. 1-8, 2022.

[2] A.A. Barakabitze, A. Ahmad, R. Mijumbi, A. Hines, “5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges”, Computer Networks, vol. 167, no. 106984, pp. 1-40, 2020.

[3] I. Ahammad, M.A. R. Khan, Z.U. Salehin, “QoS performance enhancement policy through combining fog and SDN. Simulation Modelling Practice and Theory”, vol. 109, no. 102292, pp. 1-31, 2021.

[4] S. Bolettieri, D.T. Bui, R. Bruno, “Towards end-to-end application slicing in Multi-access Edge Computing systems: Architecture discussion and proof-of-concept”, Future Generation Computer Systems, vol. 136, no. 1, pp. 110-127, 2022.

[5] N. Yarkina, L.M. Correia, D. Moltchanov, Y. Gaidamaka, K. Samouylov, “Multi-tenant resource sharing with equitable-priority-based performance isolation of slices for 5G cellular systems”, Computer Communications, vol. 188, no. 1, pp. 39-51, 2022.

[6] L.A. Phan, D.T. Nguyen, M. Lee, D.H. Park, T. Kim, “Dynamic fog-to-fog offloading in SDN-based fog

computing systems”, Future Generation Computer Systems, vol. 117, no. 1, pp. 486-497, 2021.
[7] B. Liang, M.A. Gregory, S. Li, “Multi-access Edge Computing fundamentals, services, enablers and challenges: A complete survey”, Journal of Network and Computer Applications, vol. 199, no. 103308, pp. 1-20, 2022.
[8] D. Alotaibi, V. Thayananthan, J. Yazdani, “The 5G network slicing using SDN based technology for managing network traffic”, Procedia Computer Science, vol. 194, no. 18, pp. 114-121, 2021.
[9] N. Nouri, A.A. Tadaion, “Multi Objective Optimization for Computational and Communicational Resource Allocation Based on Non-orthogonal Multiple Access, Cloud and Edge Servers Participation in 5 Generation Networks”, Tabriz Journal of Electrical Engineering, vol. 50, no. 1, pp. 451-462, 2020 (in persian).
[10] B. Yan, Q. Liu, J. Shen, D. Liang, B. Zhao, L. Ouyang, “A survey of low-latency transmission strategies in software defined networking”, Computer Science Review, vol. 40, no. 100386, pp. 1-36, 2021.
[11] K.T. Bagci, A.M. Tekalp, "Dynamic resource allocation by batch-optimization for value-added video services over sdn", IEEE Transactions on Multimedia, vol. 20, no. 11, pp. 3084–3096, 2018.
[12] Z. Latif, K. Sharif, F. Li, M.M. Karim, S. Biswas, Y. Wang, "A comprehensive survey of interface protocols for software defined networks", Journal of Network and Computer Applications", vol. 156, pp. 1-34, 2020.
[13] J. Wang, B. Zhang, T. Fan, Z. Fu, W. Zhao, "Research on multi-attribute controller for virtual data domain based on software definition network", International Journal of Distributed Sensor Networks, vol. 15, no. 7, pp. 1-11, 2019.
[14] Y.C. Chang, Y.C. Chen, T.H. Chen, J.L. Chen, S.P. Chiu, W.H. Chang, "Software-defined dynamic bandwidth management", In 21st International Conference on Advanced Communication Technology (ICACT), February 2019, Phoenix Park, PyeongChang, Korea (s), pp. 201–205.
[15] C. Liou, B. Balasubramanian, M. Rajala, S. Seetharaman, A. Sharma, S.M.S.S. Singamsetty, "Dynamic allocation of network bandwidth", United States Patent, vol. 'US 10,491,542 B2', pp. 1-21, 2019.
[16] E.L. Fernandes, E. Rojas, J. Alvarez-Horcajo, Z.L. Kis, D. Sanvito, N. Bonelli, C. Cascone, C.E. Rothenberg, "The road to bofuss: The basic openflow userspace software switch", Journal of Network and Computer Applications, vol. 165, pp. 1-24, 2020.
[17] A. Mohamed, M. Hamdan, S. Khan, A. Abdelaziz, S.F. Babiker, M. Imran, M.N. Marsono, “Software-defined networks for resource allocation in cloud computing: A survey”, Computer Networks, vol. 195, no. 108151, pp. 1-22, 2021.
[18] T. Portela, M.E. Monteiro, J.R.A. Cavalcante, J. Celestino Jr, A. Patel, "An extended software defined optical networks slicing architecture", Computer Standards & Interfaces, vol. 70, pp. 1-12, 2020.
[19] L. Ma, X. Wen, L. Wang, Z. Lu, R. Knopp, "An sdn/nfv based framework for management and deployment of service based 5g core network", China Communications, vol. 15, no. 10, pp. 86–98, 2018.
[20] k. Beigi, A.G. Delavar, “A method to improve bandwidth provision in software-defined networks”, 1st International Conference Interdisciplinary Studies in Management and Engineering, vol. 1, no. 1, pp. 1-18, 2019 (in persian).
[21] Y.W. Ma, J.L. Chen, C.C. Chang, A. Nakao, S. Yamamoto, "A novel dynamic resource adjustment architecture for virtual tenant networks in sdn", Journal of Systems and Software, vol. 143, pp. 100–115, 2018.
[22] S. Khan, F.K. Hussain, O.K. Hussain, “Guaranteeing end-to-end QoS provisioning in SOA based SDN architecture: A survey and Open Issues”, Future Generation Computer Systems, vol. 119, no. 1, pp. 176-187, 2021.
[23] S. Khorsandroo, A.G. Sánchez, A.S. Tosun, J.M. Arco, R. Doriguzzi-Corin, “Hybrid SDN evolution: A comprehensive survey of the state-of-the-art”, Computer Networks, vol. 192, no. 107981, pp. 1-40, 2021.
[24] M. Alotaibi, A. Nayak, “Linking handover delay to load balancing in SDN-based heterogeneous networks”, Computer Communications, vol. 173, no. 1, pp. 170-182, 2021.
[25] R. Chai, X. Yang, C. Du, Q. Chen, “Network cost optimization-based capacitated controller deployment for SDN”, Computer Networks, vol. 197, no. 108326, pp. 1-11, 2021.
[26] D. Santos, T. Gomes, "Joint optimization of primary and backup controller placement and availability link upgrade in SDN networks", Optical Switching and Networking, vol. 42, pp. 1-12, 2021.
[27] B. Yi, X. Wang, M. Huang, Y. Zhao, “Novel resource allocation mechanism for SDN-based data center networks”, Journal of Network and Computer Applications, vol. 155, no. 102554, pp. 1-17, 2020.
[28] A. Sethi, M. Bisht, “QoS-aware Self-optimized Reconfigurable Framework for Hyperconnected Network”, Global Transitions Proceedings, vol. 2, no. 1, pp. 18-23, 2021.
[29] B. Ahmadi, Z. Movahedi, “Stable Distributed Load Balancing between Controllers in Software Defined Networks”, Tabriz Journal of Electrical Engineering, vol. 49, no. 1, pp. 13-23, 2019 (in persian).
[30] A. Ghiasian, “Frequency scaling approach to reduce power consumption of Openflow switches”, Tabriz Journal of Electrical Engineering, vol. 49, no. 3, pp. 1273-1282, 2019 (in persian).