Scientific Research Publishing

Int. J. Communications, Network and System Sciences, 2018, 11, 53-67
http://www.scirp.org/journal/ijcns
ISSN Online: 1913-3723
ISSN Print: 1913-3715

Analysis of Dynamic Virtual Private Networks Resource Allocation Schemes

Chinenye Ajibo Augustine[1], Chukwukaelo Osuizugbe Anthony[1], Chinaeke Ogbuka Maryrose[1], Nwafor Chukwudi[2], Ikechukwu Ani Cosmas[1]

[1]Department of Electronic Engineering, University of Nigeria, Nsukka, Nigeria
[2]Department of Electrical and Electronics Engineering, Federal Polytechnic, Bida, Nigeria
Email: augustine.ajibo@unn.edu.ng, mail2chuka@gmail.com, ifeanyi.chinaeke-ogbuka@unn.edu.ng, chuksrazi@yahoo.co.uk, cosmas.ani@unn.edu.ng

How to cite this paper: Augustine, C.A., Anthony, C.O., Maryrose, C.O., Chukwudi, N. and Cosmas, I.A. (2018) Analysis of Dynamic Virtual Private Networks Resource Allocation Schemes. *Int. J. Communications, Network and System Sciences*, **11**, 53-67.
https://doi.org/10.4236/ijcns.2018.114005

Received: October 30, 2017
Accepted: April 24, 2018
Published: April 27, 2018

Abstract

The need for high performance resource allocation schemes for Virtual Private Networks (VPNs) has led to the proliferation of algorithms for VPN resource allocation. It was found that most works on VPN resource allocation focused either on admission control or link reservation on the network. Also, review of relevant literatures has revealed the need for a resource allocation/scheduling scheme whose algorithm will be able to allocate bandwidth and memory resources to different VPNS sharing the same link to the network service provider. Resources should be allocated in such a manner that utilization is optimal while VPN endpoints or customers receive services that do not undermine the service-level agreement (SLA) with the service provider. MATLAB Simulink was used to design a simulation model for analysing the VPN access network obtaining and comparing results for the link bandwidth utilization, buffer memory utilization and packet loss rate performances of the RDVNP (Robust Dynamic Virtual Network Provisioning) algorithm against the DWARF-Net (Dynamic bandwidth Allocation and guarantee on Resource Fairness) algorithm. From the results obtained, DWARF-Net algorithm's performance was better than the RDVNP's algorithm in almost all parameters tested on. On bandwidth utilization, DWARF-Net had an average channel utilization of 61.23% against RDVNP's 48.28%, on buffer utilization 42% for DWARF-Net, 41% for RDVNP and average loss rate average of 1 Packet/second for DWARF-Net against 20 Packets/second for RDVNP. From the simulation analysis and result of this work, DWARF-Net is recommended as an optimal performing algorithm for VPN resource allocation.

Keywords

ATM, DWARF-Net, MPLS, RDVNP, VPN

1. Introduction

For long, traditional Private Networks (PNs) were established by connecting Private Network sites (e.g., campuses or branch offices of enterprises) with leased lines over a Wide Area Network (WAN). Since these lines were dedicated lines, security and bandwidth guarantees were assured [1]. As enterprises and other customer's network sites proliferated and spread globally, the number of endpoints of PN's sites has spread while the endpoints got more geographically dispersed. The distance between endpoints in a Private Network is directly proportional to the fee or cost of providing the links in the private network. Thus, connecting a large number of dispersed PN sites with dedicated lines became very expensive. As a result, there was a need for a cheaper readily available alternative to the PNs. The remedy was provided by developing the Virtual Private Network (VPN) services which runs over the public network's backbone or over the public Internet. This method has been quite successful in making VPNs ubiquitous in interconnecting Private network sites. Virtual Private Network which is described as "a way to simulate a Private Network over a Public network, such as the Internet" [2], is called "Virtual" because it depends on the use of virtual connections; that is, temporary connections that have no real physical presence but consist of packets routed over various machines on the Internet on an ad hoc basis. Secure virtual connections can be created between two machines, a machine and a network, or two networks [2].

The main challenge facing the VPN has been the need to provide performance guarantees comparable to those obtained in a traditional Private Networks implemented over a Wide Area Networks (WAN) using dedicated leased-lines [1]. Traditional VPN services are offered based on two major paradigms: Overlay Virtual Private Networks where the Service Provider provides virtual point-to-point links between customer sites and Peer-to-Peer Virtual Private Networks where the Service Provider participates in the customer routing. Initially, traditional VPN implementations were built on the overlay model. In this paradigm, the Service Provider sells virtual circuits between customer sites as a replacement for dedicated point-to-point links. The overlay model has a number of drawbacks, most important of them being the need for the customer to establish point-to-point links or virtual circuits between sites. For "n" number of point to point sites, ((n) (n − 1))/2 numbers of links or virtual circuits were needed in a best-case scenario. For example, for a full-mesh connectivity between four sites, you will need a total of 6 point-to-point links or virtual circuits [3]. In order to overcome these drawbacks (particularly in Internet Protocol-IP based customer networks) and provide the customer with optimum data transport across the Service Provider backbone, another model called peer-to-peer VPN was introduced where the Service Provider actively participates in customer routing, accepting customer routes, transporting them across the Service Provider backbone and finally propagating them to other customer sites [3].

These traditional VPNs models had some major challenges namely: the in-

ability of IP to support QOS when using the peer-to-peer model. Also in the traditional VPN overlay model (*i.e.* an IP network with an ATM backbone) to make communication between "N" routers in IP network, a mesh of the order "NxN" is needed in the ATM backbone. Similarly, ATM networks are mainly used for backbones rather than at end user, converting IP to ATM cells and back can incur a huge overhead to the system. Furthermore, all the routers using the Internet Protocol within same IP-over-ATM network are adjacent to each other. If there is any topology change and update in the network occurs, the update will be sent to all routers in that network. Keeping in mind N^2 mesh within IP over ATM network, there will be N^4 topology update messages for a single topology update in the network. Consequently large computation is required with the increasing number of routers in IP over ATM networks; as such the overall scalability is decreased. Lastly as ATM network within core is capable to provide QOS parameters, yet QoS available at link layer can't be extended to IP layer as IP does not support any QoS guarantee [4].

In recent years, the Multiprotocol Label Switching (MPLS) has emerged as a worthy replacement for IP and the overlay VPNs (ATM and Frame Relay). MPLS VPN combines the best features of traditional Overlay and Peer-to-Peer VPNs; enabling the Provider Edge routers (PE) participate in customer routing, guaranteeing optimum routing between sites, easy provisioning and customers can use overlapping addresses. The PE routers carry a separate set of routes for each, making it seem like each customer is connected to a dedicated PE router. The basic aim of a VPN service provider is to improve the utilization of network resource while satisfying the user's profiled-traffic demand as contained in the Service Level Agreement (SLA), that is to say, the bandwidth reserved for VPN must be utilised optimally, while the QoS of the VPN services to the customer should be satisfied [5]. Thus, this paper emphasis on how best to effectively control the utilization of the link and guarantee the QoS for each traffic class, using the hose model to provision VPN QoS. In order to realize the objectives of this work, the remaining part of this work is organised as thus: Section II will provide an insight to MPLS network, MPLS VPN and other research works on QoS and resource allocation schemes for MPLS VPNs. While in Section III, the models for the selected VPN schemes are studied and adopted. Furthermore, in section IV simulation results are obtained, presented and discussed and finally, in Section V, conclusions are drawn from findings.

2. Literature Review

Multi-Protocol Label Switching started out as tag switching by Cisco Inc., it was developed to solve the problems facing the major WAN technologies, IP and ATM [6]. The challenges of IP networks were due to its destination based routing which required each router in the link to independently look up for the "best route" to route traffic to a particular destination IP address. Hence some links which were seen as having low bandwidth or more cost were avoided in favour

of higher bandwidth or less costlier links thus creating the what is called the "Fish problem" [7]. This result in frequent congestions in the over-used links while leaving other links unutilised. Thus, the only way out was to use those unutilised routes for policy based routing but this method was not scalable. ATM was introduced to solve this problem by using the virtual paths and virtual circuits, using Virtual Channel Interface and Virtual Path Interface to create switched paths for cells. ATM was optimised to carry voice and multimedia traffic with little delay, but it came at a price, each cell was 53 bytes long with 5bytes header. For very large file transfer in the same direction or trying to implement a very large WAN *i.e.* the internet through an ATM Network backbone would be problematic as scalability and the relatively large header bytes would cause a large overhead costs to the network [8]. MPLS solved these problems by attaching a short label (32 bits) to an IP packet header or the header of any datagram traffic passing through its network. Packet forwarding is then implemented based completely on the contents of the label rather than the contents of the IP address [9]. In MPLS, a label is mapped to an egress (exit) router rather than with the destination IP address of the packet. The MPLS Label is usually a small piece of information attached to a packet that tells every intermediate router to which egress edge router it must be forwarded. The core routers do not forward the packets based on the destination IP address, but only from the labels. The edge routers however do look at the destination IP address of the packet, in order to forward them in or out of the MPLS network hence the needs to run Border Gateway Protocol (BGP). Each BGP prefix on the ingress MPLS routers has a BGP next-hop IP address associated with it. This BGP next-hop IP address is an IP address of an egress MPLS router. The label that is associated with an IP packet is the label that is associated with this BGP next-hop IP address. Because every core router forwards a packet based on the attached MPLS label that is associated with the BGP next-hop IP address, each BGP next-hop IP address of an egress MPLS router must be known to all core routers. Any interior gateway routing protocol, such as OSPF or ISIS, can accomplish this task [10]. Label switching can appropriately be described as a forwarding model allowing streamlined forwarding of data by using labels to identify classes of data packets which are treated indistinguishably when forwarding [11].

Two popular models are used to describe QoS in the VPN framework, they are the "pipe" model and the "hose" model which are explicitly covered in [12] [13]. In the Pipe model (also known as the "customer pipe" model) a VPN service provider supplies a VPN customer with certain QoS guarantees for the traffic from one customer's Customer Edge router to another [14] [15] while In the hose model, VPN customers just need to specify the incoming and outgoing traffic volume of each VPN endpoint (known as ingress bandwidth and egress bandwidth) instead of between every pair of VPN endpoints.

Several researches has been carried out on MPLS-VPN networks mostly in the area of QoS provisioning, a brief review of these works is thus presented:

According to [12] [13] [14] [15] [16], in finding ways of dealing with challenge associated with uncertainty in a traffic's spatial distribution and provide enough resources to accommodate for the worst case traffic variation, two approaches were developed namely: static and dynamic approaches. While the static Provider-Pipes approach is the most simple of the static schemes, consisting of reservations at the Hose's peak rate between any two service endpoints, has very low efficiency and the over-provisioning factor, *i.e.* the bandwidth requirement has been found to increase linearly with the number of nodes in the network, thus making this approach inappropriate for practical application as scalability becomes an issue. More efficient static schemes make use of shared reservations for tunnels with a common service endpoint or, more generally, resources are shared among tunnels of the same VPN on common links anywhere in the network. Some authors proposed an algorithm to calculate multi-path topologies and compared the performance of several approximation algorithms [17]. They found that running times of these topology computation algorithms increase very quickly with the number of nodes and can be in the order of minutes for large networks. It has been shown that in order to achieve reasonably low over-provisioning factors, the computation of a tree-structured resource-sharing topology for the whole VPN using explicit routing is the only viable candidate among the statically provisioned models without multi-path routing [16]. In general, these computations require a global view of the VPN and the parameters on the respective service endpoints. A new resource management concept was proposed, it was named "the point-to-set model" [13], the major drawback of this customer-pipes model is that it requires detailed information on traffic distribution for a set of destinations and the mean and variance of the traffic fraction to each of these service endpoints. However, this specification still trades off flexibility of the customer's traffic patterns against resource efficiency of the VPN realization in the provider network. Volner *et al.* set out to develop a mathematical model for allocating VPN connections to bandwidth [18] for this model. Rakovetic *et al.* presented the strategy of Dynamic Partitioning of link bandwidth in IP networks [19]. In the Dynamic Partitioning scheme the bandwidth of each link in the network is partitioned into two parts, one for the low-priority data traffic, and one for the high-priority stream (real-time) traffic. The partitioning is defined by the partitioning parameter, which changes according to the traffic profile and intensity. A scheme for the support of QoS over VPN was described in [20]. They used a combination of Diffserv, MPLS and a dynamic resource allocation technique in order to provide a QoS-enabled VPN. They carried out dynamic resource allocation using traffic predictors. A solution was proposed to reduce the bandwidth over provisioning factor of the hose-based VPN solutions using a two-step resolution approach [21]. First, a pipe workload was obtained exactly using the user specified hose and a mathematical programming formulation. Second, a VPN solution was obtained using the pipe-based integer programming formulation of the VPN pro-

visioning problem. Similarly, a distributed bandwidth resizing algorithms was proposed for optimizing inter-VPN and intra-VPN bandwidth allocations [22]. The work was developed to increase the number of connections possible and as well as increase the utilization of the system. The link bandwidth or the bandwidth allocated to tunnels over the link was partitioned based on the user's utility. Any change in the users' requirement was followed by a linear change in the partitioning parameter (a). They hypothesised that the approach suggested in their work would enable VPN service providers satisfy more number of users with a quality of service guarantee and also at the same time improve their revenue. The Multi-commodity Flow Problem (MFP) solver was deployed to carry out bandwidth allocation [23] [24]. In other to avoid producing bottleneck links, they employed traffic predictor to ensure that any inaccuracy would cause the links not to have enough capacity and violate the linear constraints on the commodities for each link is avoided. They proposed the L-PREDEC for Forecasting Virtual Network dynamic link usage, while employing a linear predictor. Traffic predictor adjusted the link with the largest occupation (bottleneck link) by periodically monitoring the traffic rate of a user link and adjusting the reserved bandwidth based on the Forecasting made from the traffic history. While a scheme was proposed for enhancing VPN QoS using a method called "Log-infinitely Divisible Cascades" [25]. They proposed that the scheme would enable Several VPN traffic between two Provider Edge routers and of the tag stack using MPLS VPN data also they allowed different transmission to share a LSP tunnel. The VPN multiplicity reduces the signalling load and the scale of forwarding table of routers to bring high system scalability. MPLS based load balancing was proposed for VoIP flows, it was implemented with an effective flow classification technique which prioritized the Voice packets based on their flow arrival rate and bandwidth utilization using probing techniques [26]. They were differentiated as responsive, unresponsive and dataflow. Whenever the network experienced unbalanced load conditions due to link failure or system failure in the IP network, default routing policy would be overridden by multipath routing policy. The data rates of unresponsive flow were estimated and marked based on their data rates using Rainbow Fair Queuing (RFQ) mechanism. In order to perform multipath dispersion and alleviate the problem of congestion, the core router looks for congestion free paths and reroutes the flows into best multiple paths that satisfies the given QoS requirements. Otherwise, it drops some of the low priority packets from those unresponsive flows Quality of Service (QoS) requirements of next generation IP-based backbone networks for managing multiple VPN was considered for services offered by a VPN Service Providers [27]. In their paper, they proposed a programmable Tempest framework for Class of Service (CoS) Based Resource Allocation (CBRA) in Multi-Protocol Label Switching (MPLS) tunnelled VPNs. Switchlet based resource partitioning concept were used to create, build and provision multiple VPNs on demand. Furthermore, a distributed algorithm is proposed by using

the primal decomposition method. The algorithm through the coordination of the global coordinating algorithm operate in the network while through the local adjusting algorithm operate in the individual virtual private networks. A new fairness and bandwidth guarantee model for providing isolated network service for tenants in a virtual network was proposed for Dynamic Bandwidth Allocation and Guarantee for Virtualized Networks [28] [29] [30]. They used a distributed architecture to realize the bandwidth guarantee model by dynamically limiting the rate of each flow at the network edge. The experiment result exhibited a better utilization and better response under traffic burst. Stable and fair bandwidth allocation were obtained under different traffic patterns, and revealed satisfying response time in dynamic traffic scenario.

Summarily, most dynamic VPN provisioning algorithms reviewed in this section showed a propensity to work and improve the core network performances such as link setup time, link rerouting parameters, hose optimisation etc. all within the core network. A few others only looked at the traffic flow at the user-network interface especially the Provider Edge (PE) routers to obtain flow parameters to help optimise the bandwidth utilization at network providers' end. Also some Dynamic resource allocation or scheduling algorithms were only interested in scheduling access to the core network without considering the utilization of service provider's link. For the purpose of this work and considering the limitations mentioned above, the following QoS provisioning algorithms: Robust Dynamical Virtual Network Provisioning (RDVNP) and Dynamic bandwidth Allocation and guarantee on Resource Fairness (DWARF-Net) were selected to be investigated on as they were deployed to improve KPI of the network provider and achieve SLA with the customer.

3. System Model

3.1. Robust Dynamical Virtual Network Provisioning (RDVNP)

The Virtual network was represented by an undirected graph G (V, E), where V and E are the set of substrate nodes and the set of physical links, respectively. It was assumed that k number of VPNs co-existed on the substrate network.

A set of k VPN was represented by a set of virtual links, denoted by

$$E^{(k)} = \left\{ \left(s_1^{(k)}, t_1^{(k)} \right), \cdots, \left(s_{l_k}^{(k)}, t_{l_k}^{(k)} \right) \right\} \tag{1}$$

where (s, t) is a virtual link connecting node s and t, and l_k is the number of virtual links of the k-th VPN.

Let $S^{(k)}$ denote the set of nodes of the k-th VPN, By using the hose model constraints, all the virtual links connected to the node i will have an upper bound bandwidth constraint of $\beta_i^{(k)}$ for the k-th VPN. Therefore the traffic demands of the k-th VPN from the link is $d_{ij}^{(k)}$

$$\sum_{\forall(i,i)V(i,j)\in E^{(k)}} d_{ij}^{(k)} \le \beta_i^{(k)}, \quad \forall i \in S^{(k)} \tag{2}$$

$$\beta_i^{(k)} = \mu \sum_{(i,j)\in E^{(k)}} \alpha_{ij}^{(k)} \tag{3}$$

where $\alpha_{ij}^{(k)}$ is the upper bound of the traffic demand of virtual links. And μ can be adjusted from 0 to 1but was set as 0.8 for optimal performance.

If $c_l^{(k)}$ is the allocated bandwidth to link *l*.

And $v_l^{(k)}$ is the link weight of a node in the VPN.

The algorithm for the dynamic bandwidth allocation system is shown below

1) Given $v_l^{(k)}, \forall l \in E$ for each VPN

2) $v_l^{(k)} \leftarrow c_l^{(k)} - \theta_i \cdot \lambda_i^{(k)}$;

3) send $v_l^{(k)}, \forall l \in E$ to the global coordinating algorithm;

4) wait for receiving $c_l^{(k)}$, from the coordinating algorithm;

5) Receive $v_l^{(k)}, \forall l \in E$ from each VPN;

6) Solve min $\left\{\sum_k \left(c_l^{(k)} - v_l^{(k)}\right)^2 : \sum_k c_l^{(k)} \le c_l\right\}$, $\forall l \in E$;

7) For $\forall k$, send bandwidth $c_l^{(k)}, \forall l \in E$ to the *k*-th VPN.

3.2. Dynamic Bandwidth Allocation and Guarantee on Resource Fairness (DWARF-Net)

The bandwidth allocated to each VPN is guaranteed by rate limiting of other VPN connections, this algorithm allows busy VPN traffic to take more bandwidth beyond its guaranteed bandwidth by "borrowing" the underutilized bandwidth from idle VPNs, thus better bandwidth utilization can be achieved [29]. Such extra bandwidth, *i.e.*, bandwidth allocated beyond the guaranteed bandwidth, would be fairly allocated among virtual Ports or VPN connections from different tenants with the weight proportional to the guaranteed bandwidth they purchased. Although the algorithm was designed for both traffic transmitted from the sites into the network and for traffic received from the network only the algorithm for transmitted traffic was used for this work. The work was modelled to satisfy Pareto Efficiency, *i.e.* when there is enough free bandwidth, any machine sharing the bandwidth could be able to use it by exceeding its own set threshold:

$$\sum_{i\in host_k} t_{ij} \ge B \Rightarrow \sum_{i\in host_k} b_{ij} = \eta B \tag{4}$$

where η is the overall bandwidth utilization and B is total bandwidth of the link

$$\forall i, j \sum_{j\neq i} t_{ij} \ge A_i^t \Rightarrow \sum_{j\neq i} b_{ij} \ge A_i^t \tag{5}$$

$$\sum_{i\in host_k}^{n} A_i^t \le B_k \tag{6}$$

$$\forall i, \sum_{j\neq i} G_{ij} \le A_i^t \tag{7}$$

T*i* represents the traffic sending request of VPN *i*, based on the model above,

$$T_i = \sum_{j\neq i} b_{ij} \tag{8}$$

3.3. Proposed Network Architecture

For the sake of this work some assumptions are made for the network architecture;

1) There are three different VPNS connected to the service provider edge router, which has a leased link through the network provider's network.

2) That these VPNS are connected to their other site at another end through a provider edge router as seen in Figure 1.

3) The VPNs are not just work stations, but can contain several workstations and servers for database, mail, ftp, HTTP and voice traffic. But for the sake of this research, only one provider edge router will be considered as MPLS creates links or LSPs in one direction.

The above network architecture was implemented in MATLAB Simulink, the model created focused on the Edge router at node 2.

3.4. Proposed Physical Model (Figure 2)

The physical model explains the interface at which the resource allocation schemes being compared are deployed. A brief description of each block is provided below:

1) Admission control: this interface selects which node is served at a particular time (t).

2) Packet classifier: packets are distinguished in order of their priority and sent to their different queues.

3) Buffers: the buffers queue the packets according to their different priorities, Note higher priority packets have shorter queues while lower priority packets have longer queues.

4) Metering phase: Calculates the difference between the incoming packets and the served packets and also determines the bandwidth to be reserved in the core network.

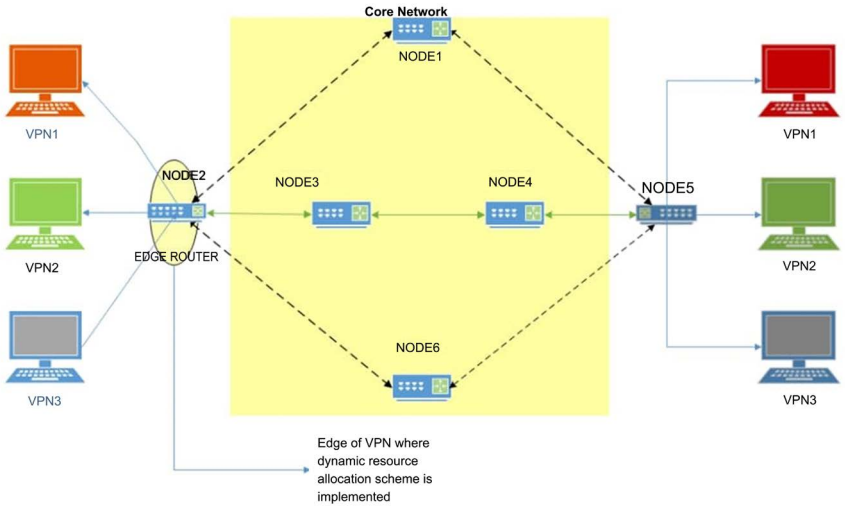

Figure 1. Proposed Physical Network Architecture.

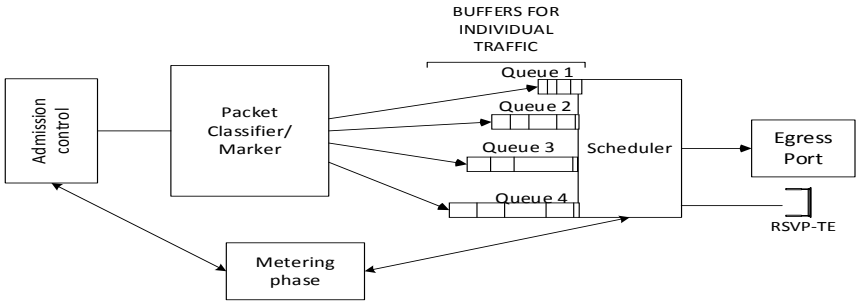

Figure 2. VPN Scheduler Architecture.

5) Scheduler: this is where the algorithm for the schemes being compared are implemented in the form of MATLAB codes.

6) Egress port is the external channel to the core network.

4. Simulation Result and Analysis

Twelve simulations were carried out with 12 different mean (*i.e.* intergeneration time for time based entity generator). The needed parameters were sent to Matlab workspace from where the mean of the generated parameters were obtained for each intergeneration time used.

Figure 3 shows the relationship between intergeneration time and Packet generation rate. Intergeneration time and packet generation exhibit an inverse relationship, hence the lower the intergeneration time the higher the packet rate from the generators.

The parameters needed were Link utilization which is already available from the server block, packet arrival rate from the blocks, buffer utilization which was calculated thus

$$\text{Buffer Utilization} = \frac{\text{Number of packet in buffer}}{\text{Total buffer size}} * 100$$

While loss rate was derived from the Simulink blocks by collating the number of dropped or timed out packets, dividing the number of dropped packets by the simulation time and sending the values to workspace. The increase in Packet generation rate as seen in **Figure 3** is seen to cause an increase in the utilization of both RDVNP and DWARF-Net as seen in **Figure 4**.

Figure 4 shows that DWARF-NET algorithm allowed link utilization to peak faster than the RDVNP algorithm, also this performance was maintained throughout the simulation, the DWARF-NET algorithm produced an average utilization of 61.23% a score that was larger than that of the Robust Dynamical Virtual Network Provisioning (RDVNP) algorithm's 48.28% by a factor of 12.94% obtained by comparing the average result of the two schemes. The standard deviation from the result on Channel Bandwidth utilization equalled 40% for DWARF-NET, meaning the results were widely spread for the different intergeneration times used, while the standard deviation was 35% for the RDVNP algorithm meaning it had a lower spread and less response to varying traffic being generated. This result puts DWARF-Net on a better performance in terms

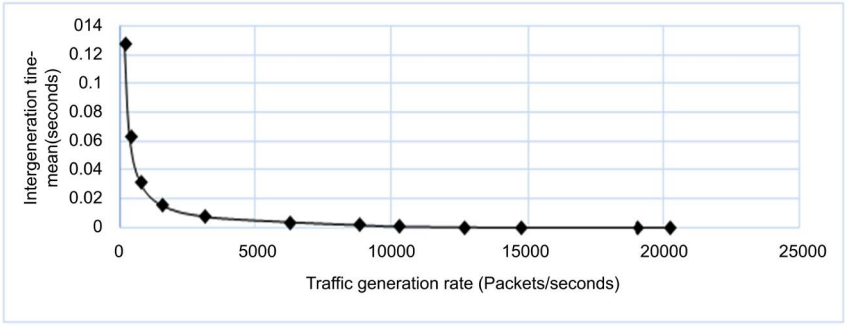

Figure 3. Packet generation against Intergeneration time.

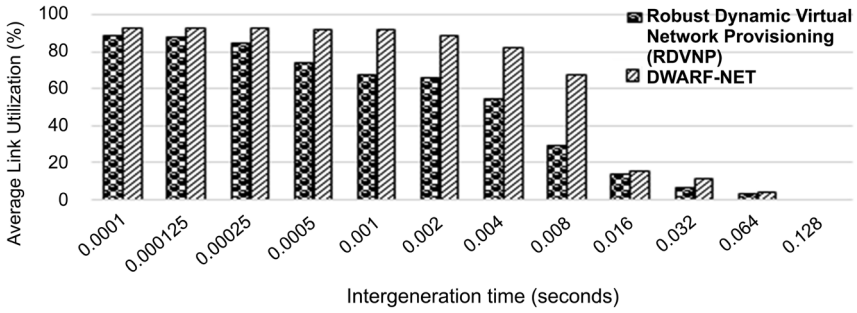

Figure 4. Link Bandwidth Utilization Chart of RDVNP against DWARF-Net.

of bandwidth utilisation than the RDVNP algorithm.

Similarly, graphical results obtained while investigating buffer utilization for the two algorithms as presented in Figure 5, shows that for most values of lower intergeneration time or periods of during which high packet traffic flooded the system, RDVNP utilized the buffer optimally and performed better than the DWARF-NET algorithm in terms of buffer utility, but at lower traffic the DWARF-NET had a better utilization of the buffer. Comparing the average score of values generated for the two schemes showed that the average of the utilization of the buffer by DWARF-Net algorithm was 42% which was superior to the average utilization obtained from RDVNP which was 41% utilization, therefore averagely while the RDVNP algorithm had a better utilization at higher traffic, DWARF-NET algorithm had the overall best average performance. It is safe to infer that RDVNP sacrifices link utilization for better buffer utilization when packets flood the system. While the DWARF-Net algorithm is designed to reduce the system waiting time and optimise service rate for high traffic systems. Its buffer utilization also tends to suggest that it stabilizes the network and controls erratic traffic behaviour. The RDVNP scheme produces a standard deviation of 42% against DWARF-NET's standard deviation of 35%. The results obtained from the standard deviation showed that the RDVNP is quite unstable in its buffer utilization as it had a wider range of values above the average score for buffer utilization than the DWARF-NET algorithm which had a smaller range of values which occurred mainly below the mean or average score for the buffer utilization obtained.

Also Figure 6 shows the results for 12 simulations at different intergeneration

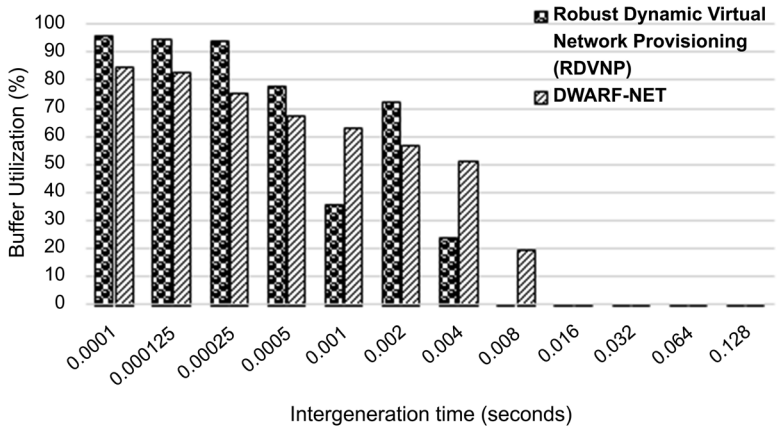

Figure 5. Buffer Utilization Chart of RDVNP against DWARF-Net.

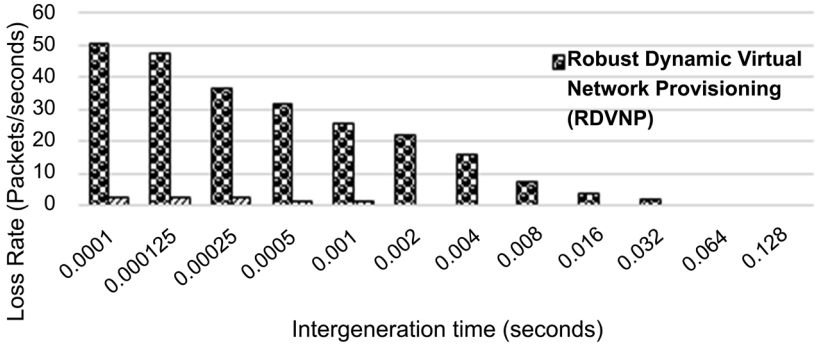

Figure 6. Loss Rate for RDVNP against DWARF-Net.

time, the result shows that the RDVNP algorithm has a very high loss rate compared to DWARF-Net. The RDVNP produced an average loss rate of 20 Packets/seconds which is 95% more than the average loss rate of 1.0 Packets/seconds recorded for DWARF-Net. The standard deviation for the RDVNP loss rate is at 18 packets/seconds, with a corresponding spread between 2 and 38 Packets/second making it more unstable than the DWARF-Net algorithm which has a standard deviation of 1 Packet/second and a spread between 0 and 2packet/second. The higher channel utilization and buffer utilization produced a very low loss rate for DWARF-Net and the reverse was the case for RDVNP. The stabilizing effect and efficient traffic scheduling of the DWARF-NET algorithm is seen to contribute to its low loss rate throughout the simulation when compared to the RDVNP algorithm. RDVNPs high loss rate despite its high buffer utilization suggests that it does not efficiently allocate resources to time dependent packets, meaning that switching packets isn't intelligent and efficient enough to forward time dependent traffic like voice at the expense of non-time dependent traffic like Best effort traffic.

Table 1 shows the formula descriptor of the DWARF-NET model, while Table 2 shows the statistical data obtained from the charts in Figures 4-6. On inspection of the table, it is clear that the DWARF-NET scheme outperformed the RDVNP scheme using the statistical analysis used and presented in Table 2.

Table 1. DWARF-NET model formula descriptor.

t_{ij}	Traffic sending request from VPN i to j
U_i	Upper bound bandwidth for VPN i
B_k	Total bandwidth shared in the host k
A_i^t	Tx Access bandwidth guarantee for VPN i
G_{ij}	Pair Bandwidth guarantee from i to j
T_i	Traffic sending request of VPN i
t_{ij}	Traffic sending request from VPN i to j
U_i	Upper bound bandwidth for VPN i
B_k	Total bandwidth shared in the host k
A_i^t	Tx Access bandwidth guarantee for VPN i
G_{ij}	Pair Bandwidth guarantee from i to j
T_i	Traffic sending request of VPN i

Table 2. Statistical data from simulation results.

	Channel utilization (%)		Buffer utilization (%)		Loss rate (packets/seconds)	
	Average (statistical mean)	Standard deviation	Average (statistical mean)	Standard deviation	Average (statistical mean)	Standard deviation
Rdvnp	48.28	35	41	42	20	18
Dwarf-net	61.23	40	42	35	1	1

5. Conclusion

As start-ups blossom into enterprises, there is need to have dispersed sites connected together in private networks to enable secure communication between staffs who work remotely and the office, different sites of the same enterprise (site to site) and extranet based connections *i.e.* some third party connecting to the network. Due to the outrageous cost of building networks ground-up, network administrators must be able to intelligently use public network resources while preserving the security of data as well as ensure prudent management of resources at the network edge. As the trend towards Next Generation Networks (NGN) and Software Defined Network SDN, there is need to have scheduling schemes whose metrics points towards optimal performances for provisioning VPN QoS. Therefore, this paper has succeeded in providing a simulation model that researchers can use in analysing Virtual Private Networks. The model was used to investigate the performances of two VPN resource scheduling and allocation schemes by varying the traffic generated and obtaining results with respect to Link utilization, Buffer utilization and packet loss rate. In terms of link utilization, an increment was observed with more packet traffic in the network but this limited scheduling algorithm from reaching optimal level. This observed behaviour can be attributed to the delay in executing the algorithm. But the

DWARF-Net algorithm provided more utilization against all other algorithms. Similarly, the buffer utilization increased with increase in packet generation. Also the loss rate increased with packet generation, but also lower losses were experienced in DWARF-Net due to the higher utilization of the link and buffer.

References

[1] Augustine, A., Chukwudi, I. and Cosmas, A. (2015) Performance Evaluation of Enterprise-Wide Network that It's Backbone Is Based on Leased Trunk. *International Journal of Communications, Network and System Sciences*, **8**, 399-407. https://doi.org/10.4236/ijcns.2015.810037

[2] Scott, C., Wolfe, P., Erwin, M. and Tunnel, A. (1998) Virtual Private Networks. 2nd Edition, O'Reilly, Sebastopol.

[3] Augustine, A., Chukwudi, I. and Cosmas, A. (2015) Dynamic Resource Allocation Scheme for an ATM Based Enterprise-Wide Network. *International Journal of Scientific & Engineering Research*, **6**, No. 11. http://www.ijser.org

[4] Ahmed, M. and Basit, A. (2014) Implementation of Traffic Engineering and Addressing QoS in MPLS VPN Based IP Backbone. *International Journal of Computer Science and Telecommunication*, **5**, No. 6.

[5] Zou, Y., Mi, Z. and Meng, X. (2006) A Genetic Algorithm for Optimization of Bandwidth Assignment in Hose-Modeled VPN. *Lecture Notes on Computer Science,* No. 60472105, 315-323. https://doi.org/10.1007/978-3-540-37275-2_41

[6] Osuagwu, D.H.O., Ajibo, A.C., Ugwuanyi, S.O., Nwachi-Ikpo, J. and Ani, C.I. (2017) Dynamic Bandwidth Scheduling for WCDMA Uplink Transmission. *International Journal of Scientific & Engineering Research*, **8**, No. 2.

[7] Ali, S. and Rana, B.Z. (2011) OPNET Analysis of VoIP over MPLS VPN with IP QoS. Master's Thesis, Blekinge Institute of Technology, Blekinge.

[8] Stallings, W. (2001) MPLS—The Internet Protocol Journal—Volume 4, Number 3. http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_4-3/mpls.html

[9] Davie, B.S. and Farrel, A. (2008) MPLS: Next Steps. 2nd Edition, Morgan Kaufmann Publishers, Burlington.

[10] Forouzan, B.A. (2007) Data Communication and Networking. 4th Edition, McGraw-Hill, New York.

[11] Xiao, X., Hannan, A., Bailey, B. and Ni, L.M. (2002) Traffic Engineering with MPLS in the Internet. *IEEE Network*, **14**, 1-14.

[12] Ben-Ameur, W. and KERIVIN, H. (2005) Routing of Uncertain Traffic Demands. *Springer Science*, No. 6, 283-313.

[13] Ma, Z. (2007) Multipath Resource Management in Overlay Networks Multipath Resource Management in Overlay Networks. Dissertation Director , Arvind Krishnamurthy.

[14] Carugi, M. and McDysan, D. (2005) Service Requirements for Layer 3 Provider Provisioned Virtual Private Networks (PPVPNs). RFC 4031 Network Working Group.

[15] Liu, Y.L. and Chin, Y.T. (2010) Traffic Engineering for Provisioning VPNs with Time-Varying Bandwidth Requirements. 2010 *International Conference on Electronics and Information Engineering* (*ICEIE*), **2**, 309-313. https://doi.org/10.1109/ICEIE.2010.5559788

[16] Mariz, D., Kelner, J., Sadok, D. and Kamienski, C.A. (2005) The Accurate Hose

Model for VPN Provisioning. *XXIII Simpósio Brasileiro de Redes de Computadores.*

[17] Liu, Y., Sun, Y.S. and Chen, M.C. (2006) MTRA : An On-Line Hose-Model VPN Provisioning Algorithm. *Telecommunication Systems*, **31**, 379-398.

[18] Wei, D. and Ansari, N. (2004) Implementing Fair Bandwidth Allocation Schemes in Hose-Modelled VPN. *IEE Proceedings-Communications*, **151**, 521-528. ttps://doi.org/10.1049/ip-com:20040840

[19] Alpar, J., Istvan, S. and Aron, S. (2003) On Bandwidth Efficiency of the Hose Resource Management Model in Virtual Private Networks. 22*nd Annual Joint Conference of the IEEE Computer and Communications*, **1**, 386-395.

[20] Rikli, N. (2011) Evaluation of End-to-End Quality of Service over VPN Networks through Various Priority Mechanisms. *The* 6*th International Conference on Digital Telecommunications*, Beirut, 145-149.

[21] Chu, J. and Lea, C. (2007) A Restorable MPLS-Based Hose-Model VPN Network. *Computer Networks*, **51**, 4836-4848.

[22] Dong, W. and Nirwan, A. (2004) Implementing Fair Bandwidth Allocation Schemes in Hose-Modelled VPN. *IEE Proceedings-Communications*, **151**, 521-528.

[23] Byun, H. and Lee, M. (2007) Extensions to P2MP RSVP-TE for VPN-Specific State Provisioning with Fair Resource Sharing. *Computer Communications*, **30**, 3736-3745. https://doi.org/10.1016/j.comcom.2007.09.007

[24] Christian, M., Dotaro, E. and Papadimitriou, D. (2006) A Practical Approach to VPN Resource Management using a Dynamic Hose Model. 2006 2*nd Conference on Next Generation Internet Design and Engineering*, Valencia, 3-5 April 2006, 147-153.

[25] Lim, L.K., Gao, J., Ng, T.S.E., Chandra, P.R., Steenkiste, P. and Zhang, H. (2001) Customizable Virtual Private Network Service with QoS. *Computer Networks*, **36**, 137-151.

[26] Altın, A., Amaldi, E., Belotti, P. and Pınar, M.C. (2004) Virtual Private Network Design under Traffic Uncertainty. *Electronic Notes in Discrete Mathematics*, **17**, 19-22.

[27] Bai, H., Gu, F., Crichignoi, J., Khan, S. and Ghani, N. (2014) Virtual Network Scheduling Design. 2014 *IEEE* 3*rd International Conference on Cloud Networking* (*CloudNet*), Luxembourg, 8-10 October 2014, 362-367. https://doi.org/10.1109/CloudNet.2014.6969022

[28] Min, Z., Chunming, W.U., Yue, H., Qiang, Y., Bin, W. and Ming, J. (2013) Robust Dynamical Virtual Network Provisioning. *Chinese Journal of Electronics*, **22**, 151-154.

[29] Wang, J. (2014) Dynamic Bandwidth Allocation & Guarantee for Virtualized Networks in Cloud. 2013 9*th International Conference on Information, Communications and Signal Processing* (*ICICS*), Tainan, 10-13 December 2013, 1-5.

[30] Tan, L., Yang, P., Zhang, W. and Ge, F. (2012) On Utility-Optimised Router-Level Bandwidth Allocation. *Transactions on Emerging Telecommunications Technologies*, **24**, 303-316.