International Journal of Scientific and Technical Research in Engineering (IJSTRE)
www.ijstre.com Volume 8 Issue 6 ǁ Nov-Dec 2023.
ISSN: 2581-9941.

Improving Resource Management in Virtual Private Network using Modified Dynamic Hose Model

Okorogu, V.N.[1], Okafor, C. S.[2*], Egbonwonu, E.L.[3]
[1,2](Department of Electronics & Computer Engineering / Nnamdi Azikiwe University, Awka, Nigeria)
[3](Department of Electrical/Electronic Engineering / Enugu State University of Science and Technology, Enugu, Nigeria)
*Corresponding Author: cs.okafor@unizik.edu.ng

Abstract : *The goal of this paper was to employ a modified dynamic hose model to improve resource management in virtual private networks (VPNs). Inadequate network resources have presented a significant obstacle to Nigeria's internet business expansion. Allocate resources in a way that maximizes usage and ensures that VPN endpoints and customers receive services that uphold the Service-Level Agreement (SLA) that was agreed with the service providers. For the purpose of managing data flow, a modified dynamic hose algorithm is modeled in this study. The data for transmission was modeled, and the VPN under investigation was described. Next, using MATLAB, an algorithm for a modified dynamic hose model was created to manage different traffic rates. According to the results of the network characterization, changes in window and packet sizes have an impact on a VPN's throughput. For example, increasing the window size from 50 to 100 kb resulted in a 47% improvement in throughput, which went from 15 for the Conventional Hose Model to 28.3 for the Modified Dynamic Hose Model. The throughput of a VPN is also impacted by variations in window and packet sizes. For example, increasing the window size from 10 kb to 50 kb resulted in a maximum throughput of 3.01 for the Conventional Model, compared to 15 for the Modified Dynamic Hose Model, which represents an improvement of 79.93%. In contrast to the Conventional Hose Model, the Modified Dynamic Hose Model algorithm decides whether to queue or drop a certain packet, increasing the virtual private network throughput and capacity usage.*

Keywords – *Conventional Hose Model, Dynamic Hose Model, Resource Management, Throughput, Virtual Private Network.*

I. INTRODUCTION

While virtual private network services have long been available in a variety of formats, they have recently drawn a lot of interest from the Internet Protocol (IP), frame-relay, MPLS, and ATM networking communities [1]. VPNs should, at minimum, offer a comparable service because customers are likely to utilize them in place of networks built using private lines. Significant advancements in IP security technology allow us to enhance the security and privacy offered by current VPN service providers that rely on frame-relay or private lines. Routing protocols, tunneling, and group membership have been the main topics of other IP-based VPN research [1]. Issues with VPN resource management have received far less attention. But in order to handle several mission-critical operations, a VPN service needs to guarantee performance with Service Level Agreements (SLAs) in place. Private lines guarantee bandwidth, loss, and delay characteristics while isolating a VPN's performance from other flows. Similar performance guarantees must be provided by a VPN service. The performance concerns with VPNs were the main topic of this article.

The number of endpoints per VPN is expanding, and it's getting harder to predict endpoint communication patterns as a result of security advancements and the overwhelming acceptance of IP networking technologies. We anticipate that consumers won't be able to predict loads between pairs of endpoints, or won't be willing to do so. Similarly, utilizing the Conventional technique to express QoS requirements point-to-point will become more and more challenging. We refer to this approach as the Hose

Model, which functions as a performance abstraction (i.e., how a provider thinks of a VPN) and a VPN service interface (i.e., how a client thinks of a VPN). Performance assurances are provided by a Hose for both traffic to and from a certain endpoint within the VPN, as well as from that endpoint to the set of all other endpoints within the VPN. The Hose serves as the client's interface with the network and can be thought of as their "link" to the network. The customer can put traffic into the network using the Hose service interface without having to forecast point-to-point volumes. Even though the Hose Model offers Customers more straightforward and adaptable SLAs, it seems to pose a more difficult resource management issue for the Provider. The temporal variation in the traffic between the two sites is unclear under the Conventional point-to-point Model for establishing QoS. Additionally, there is geographical uncertainty under the Hose Model, or uncertainty regarding traffic sinks. We create techniques that enable providers to leverage the Hose Model and obtain notable multiplexing advantages in the network by using signals to dynamically scale the network's capacity and the Hose. This helps them deal with these uncertainties. A point-to-cloud VPN service level assurance is called a Hose. In essence, the basic design problem—that is, the amount of capacity required to support the Hoses—is examined in this study. Specifically, we want to know how much the Provider has to pay to supply enough capacity to handle traffic whose matrix isn't entirely known. We conduct several trace driven experiments to assess the proposed Hose VPN service model. Specifically, we demonstrate that when the network is able to take use of the Hose Model, large multiplexing advantages may be realized for both the provider and the customer. For these investigations, two sets of traces were employed. Traces of voice traffic from the AT&T backbone network were the first. Data traffic traces from a sizable corporate backbone network made up the second. "The totality of features and characteristics of a product or service that bears its ability to satisfy stated or implied needs" is how the International Standard Organization (ISO) defines "quality" [2]. The ISO also seeks to standardize and make sense of the language related to quality. In general, when we talk about "quality," we consider the service from the viewpoint of the users, which includes an end-to-end view as depicted in Figure 1. This perspective, however, takes into account the aggregate impact of individual performances, including those of the network, the terminal, and the customer support process. Considering the aforementioned, it is necessary to clarify a number of concepts and standards in this ecosystem that are both objective (user perceptions) and subjective (other factors).

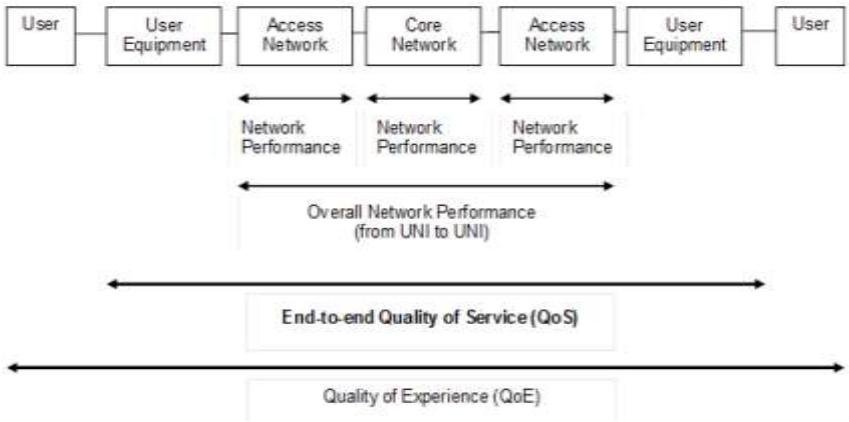

Figure 1: End-to-End Quality Experience [3]

We refer to Quality of Experience (QoE), as perceived by the user, as a subjective aspect of the service. Each user's perspective differs depending on the terminal, context, and quality of service [3].

The International Telecommunications Union (ITU) indicates that there are four different views to examine "quality" [4], all the way from the provider's perspective to the customer's, at a separate analysis level and through its recommendation G.1000. This all-encompassing vision aids in conceptualizing the two aspects of QoS that have been discussed [5].

1. The level of excellence that a client expects from a certain service (their demands)
2. The perceived (experienced) quality standard that a client claims to have

3. The level of quality that the service provider claims, or anticipates, to provide in accordance with network planning

4. The level of quality that the service has actually acquired.

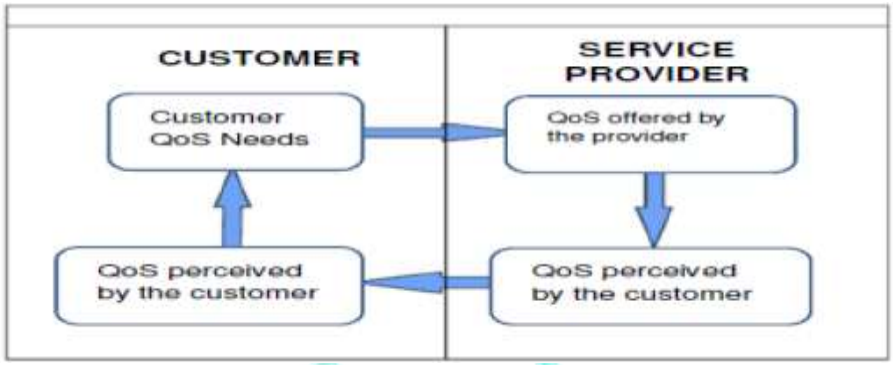

Figure 2: Four Perspectives on Quality as Defined ITU [5].

An internet service provider (ISP) can only manage the quality they anticipate providing, which is decided by design specifications and network characteristics, as we have already discussed. On the other hand, the quality level that is reached or attained is evaluated after a certain amount of time and is dependent on certain factors, such behavioral and meteorological factors, over which the ISP has limited control. This ITU perspective takes into account all factors that affect the quality that is provided and achieved by the provider, taking into account the demands and perceptions of the client [5]. With an emphasis on the provider, it is possible to comprehend and examine the objective component of QoS at several levels, including processes, access networks, core networks, terminal equipment and the performance of Wireless Sensor Networks in an energy-efficient test-bed to mitigate the undesired consequence of increased energy consumption [6]. In the field of computer networking and other packet-switched telecommunication networks, and in traffic engineering, the term QoS refers to resource reservation control mechanisms rather than achieved service quality [7], and for optimization of data throughput that improves traffic management, there is need for effective capacity and effective bandwidth requirements [8]. Although the terms "quality of service" and "quality of network performance" are sometimes used interchangeably, they have an inherent link. The link between the two ideas is depicted in the following figure:

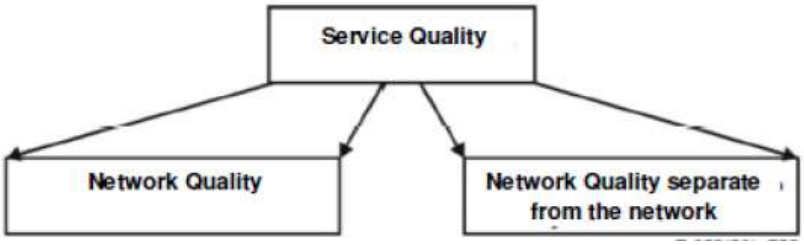

Figure 3: Relationship between Quality of Network Performance and Quality of Service [5].

Technical factors that are objectively measured and characterize a portion of the network's performance make up the quality of the network's performance. Throughput rates, latency, changeable delay, etc. are a few of these. Conversely, network-independent factors include, but are not limited to, turnaround times for processes and repairs. Depending on the type of service, these have varying weights in terms of QoS. Taking the aforementioned into consideration, the research project focuses on the quality of network performance that ISPs can control, or the quality they anticipate providing on their network, which is mostly defined by factors that rely on the access and core networks.

II. MODEL FOR RESOURCE MANAGEMENT IN VIRTUAL PRIVATE NETWORKS

According to a research study [9] titled "A Flexible Model for Resource Management in Virtual Private Networks," VPN usage is rapidly increasing as IP technologies that offer enormous capacity and the capacity to create dynamic, secure associations between endpoints emerge. There are more endpoints per VPN, and it's getting harder to predict how they will communicate with one another. Users, as a result, expect the network to support any traffic matrix, provided that the traffic to the endpoints does not exceed the rates of the corresponding ingress and egress links. Users also anticipate consistent, dynamic connectivity between endpoints. To offer the necessary performance abstraction, they suggest a brand-new service interface they call a hose. A hose reduces the amount of state information a customer must maintain when compared to the traditional point-to-point (or customer-pipe) model of operating system management. Hoses are defined by the aggregate traffic to and from one endpoint in the VPN to the set of other endpoints in the VPN, as well as by the performance guarantee that goes along with it. However, it would seem that hoses would make the already challenging issue of resource management to sustain an operating system even more problematic. They take into account both traditional statistical multiplexing strategies and a novel resizing technique based on online measurements to manage network resources in the face of this heightened uncertainty. They employ trace-driven simulations, utilizing traffic from a sizable corporate data network and AT&T's voice network, to investigate these performance concerns. The customer discovered that there are notable multiplexing advantages when traffic is aggregated at the hose level. From the standpoint of the provider, we discover that statistical multiplexing and resizing approaches give substantial improvements over the traditional alternative of a mesh of statically sized customer-pipes between endpoints by successfully handling traffic uncertainties. Even while it seems like hoses create more resource management concerns for the provider, statistical multiplexing can help with these issues in further studies.

1) Resource Management for Virtual Private Networks

In a study titled "Resource Management for Virtual Private Networks" [10], the authors believe that provider-managed solutions based on RFC 2547 are a popular option for layer-3VPNs, and that the Hose Model—which provides customers with a "hose" of a certain contracted bandwidth—has become a common and straightforward service specification. As VPNs grow in size and number, and as customer traffic patterns become more unpredictable, providers face new challenges in effectively provisioning and capacity planning for these networks while meeting customer service level agreements (SLAs). They propose a number of strategies that can be employed to assist the provider in creating an adaptively provisioned network. These methods entail the continuous processing of measurement data, the construction of conclusions about VPN attributes, and the application of these conclusions to adaptive resource delivery. With the use of an existing SNMP-based measurement infrastructure from a major IP/VPN service provider, they have proven the viability of such provisioning and have created scalable ways to infer VPN properties that are crucial for provisioning jobs. Their analysis of the measurement data produced some intriguing new findings about the composition and characteristics of VPNs. In their article, they described an adaptive provisioning architecture that enables providers to successfully handle the changing nature of customer traffic, building on the experience with assessing VPN characteristics. Their examination of traffic matrices enabled them to make significant inferences regarding the temporal and spatial properties of VPNs. They discovered that a sizable portion of the most common configurations were Hub/Spoke VPNs. The customer traffic temporal trends exhibit consistency over many weeks to a month, indicating that measurement-based learning of traffic features is feasible. There is a research gap since their research was not supported by simulation work. The Multi-commodity Flow Problem (MFP) solver was used to allocate bandwidth, according to their work on a realistic method to VPN resource management using the Dynamic Hose Model [11]. They used a traffic predictor to make sure that any errors would prevent the links from having adequate capacity and from violating the linear limitations on the commodities for each link, preventing the creation of bottleneck linkages. They employed a linear predictor and proposed the L-PREDEC for projecting dynamic link utilization in virtual networks. By regularly monitoring a user link's traffic rate and modifying the allotted bandwidth based on forecasts drawn from traffic history, the traffic predictor was able to alter the link with the highest occupation (the bottleneck link).

a) An On-Line Hose-Model VPN Provisioning Algorithm

A multi-path topology calculation approach was presented by certain authors [12], who also conducted a performance comparison of various approximation algorithms. They discovered that the number of nodes rapidly increases the running periods of various topology computation techniques, which can reach minutes for very large networks. It has been demonstrated that the only feasible option among the statically provided models without multi-path routing is the computation of a tree-structured resource-sharing topology for the entire VPN using explicit routing in order to obtain reasonably low over-provisioning factors. These calculations typically call for a global perspective of the VPN as well as the settings on the corresponding service endpoints. A novel idea for resource management was put forth and given the moniker "the point-to-set model." This customer-pipes model's main flaw is that it needs precise data on the distribution of traffic to a set of destinations, as well as the mean and variation of the traffic percentage to each of these service endpoints. This standard, however, nevertheless compromises the resource efficiency of the VPN realization in the provider network for the flexibility of the customer's traffic patterns. A mathematical model for assigning VPN connections to bandwidth for this model was developed and the approach of dynamically partitioning link bandwidth in IP networks was provided in Implementing Fair Bandwidth Allocation Schemes in Hose-Modeled VPN [13]. Each network link's bandwidth is divided into two sections under the dynamic partitioning scheme: one for high-priority stream (real-time) traffic and one for low-priority data traffic. The partitioning parameter, which varies according on the traffic profile and intensity, defines the partitioning [14].

b) Dynamic Bandwidth Allocation and Guarantee for Virtualized Networks in Cloud

For Dynamic Bandwidth Allocation and Guarantee for Virtualized Networks, a new fairness and bandwidth guarantee model was presented for offering tenants in a virtual network isolated network service [15]. They dynamically limited the pace of each flow at the network edge in order to implement the bandwidth guarantee concept using a distributed architecture. The results of the trial showed improved usage and response during spikes in traffic. In a dynamic traffic scenario, stable and equitable bandwidth allocation was achieved under various traffic patterns, resulting in a satisfactory response time.

The implementation of a quality-of-service (QoS)-enabled, Internet-based VPN management system is described in a research article [16] named "Virtual Private Network and Quality of Service Management Implementation." Though the paper also outlines an implementation that enables the transfer of fine-grained Integrated Services QoS methods to Differentiated Services, the system's QoS mechanisms are centered on scalable Differentiated Services. The Internet Protocol security architecture serves as the foundation for the VPN management system's security features (IPSec). The benefit of this idea is that RDG1's local queuing mechanism can be further expanded to include the DiffServ cloud. An excellent interface exists between the daemon and the queuing system (which is based on the programs/libs tc and ip) in the RSVP software that served as the foundation for their implementation. Thus, adding a third layer for the BB interface between the queue system and the daemon is rather simple. Thus, certain functions that query the Bandwidth Broker are called each time local resources are reserved, released, or altered. Accordingly, a reservation is only considered effective if both the BB and the local traffic control system approve of it. Keep in mind that they employ QoS-VPN ISB as their bandwidth broker. The RDG can leverage the reservation methods and connect to the broker using a dedicated interface. Consequently, another benefit is that the extension is completely transparent, meaning that no RSVP user outside of the ISB will need to make any changes or even take into account that a DiffServ mapping occurred.

Additionally, we provided design options for the RSVP to DiffServ gateway (RDG) implementation. Fine-grained "IntServ" resource reservations (RSVP) are transparently translated to the coarse-grained service broker reservation mechanism (DiffServ) through the use of the gateway. After certain reservations, the RDG contacts the service broker on its own. As a result, end users utilizing IntServ apps can obtain the proper QoS without ever having to get in touch with the service broker. The RSVP daemon of DSR is in charge of admissions management. Therefore, it is not possible to configure something in depth or handle a particular flow.

III. DESIGN

This study evaluated the network's resource management and characterized virtual private networks. A test bed was put up to track traffic and bandwidth usage on the network during the network characterization. Iperf was the traffic creation and monitoring technology used. Throughput, expressed in Mbps, is the experiment's metric. Large files were also used to evaluate the workloads across all VPN servers.

Following the VPN's characterization, a virtual private network model was created and run. Following the development of the Modified Hose Model's algorithm, the Modified Dynamic Hose Model was created.

1) Develop Algorithm for Modified Dynamic Hose Model to Handle Varying Traffic Rates

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:

$$E^{(k)} = \left\{\left(s_1^{(k)}, t_1^{(k)}\right), \ldots, \left(s_{lk}^{(k)}, t_{lk}^{(k)}\right)\right\} \quad 3.1$$

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 dij $^{(k)}$:

$$\sum_{\forall(i,i)\vee(i,j)\epsilon E^{(k)}} d_{il}^{(k)} \leq \beta_i^{(k)}, \qquad \forall i \in S^{(k)} \quad 3.2$$

$$\beta_i^{(k)} = \mu \sum_{(i,j)\in E^{(k)}} \alpha_{ij}^{(k)} \quad 3.3$$

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 $cl^{(k)}$is the allocated bandwidth to link l
And $vl^{(k)}$is the link weight of a Node in the VPN.
The algorithm for the dynamic bandwidth allocation system is shown below

1. Given $c_l^{(k)}, \forall l \epsilon E$ for each VPN
2. $v_l^{(k)} \leftarrow c_l^{(k)} - \theta_i \cdot \lambda_i^{(k)};$
3. Send $v_l^{(k)}, \forall l \epsilon E$, to the global coordinating algorithm;
4. Wait for receiving $c_l^{(k)}$, from the coordinating algorithm;
5. Receive $v_l^{(k)} \forall l \epsilon E$, from each VPN
6. Solve min

$$\left\{\sum_k (c_l^{(k)} - v_l^{(k)})^2 : \sum_k (c_l^{(k)} \leq c_l\right\}, \forall l \in E$$

7. For$^{\forall k, sendbandwidth c_i^{(k)}, \ \forall l \in E, tothek-thVPN}$

This technique enables busy VPN traffic to consume more bandwidth beyond its provided bandwidth by "borrowing" the underutilized bandwidth from idle VPNs, resulting in greater bandwidth utilization. The bandwidth allocated to each VPN is ensured by rate limitation of other VPN connections. Overage bandwidth, or bandwidth allotted above the provided bandwidth, would be equitably distributed among Virtual Ports or VPN connections belonging to various tenants, with a weight based on the guaranteed bandwidth each of them paid for. For this work, just the algorithm for transmitted traffic was utilized, even though the method was

intended to handle both traffic received from the Network and traffic transmitted from the sites into the Network. The work was designed to meet Pareto Efficiency, which states that each computer sharing bandwidth may use it by surpassing its own service level agreement.

$$\sum_{i \in hostk} t_{ij} \geq B \Rightarrow \sum_{i \in hostk} b_{ij} = \eta B \quad 3.4$$

Where ηis the overall bandwidth utilization and B is total bandwidth of the link

$$\forall i,j \sum_{j \neq i} t_{ij} \geq A_i^t \Rightarrow \sum_{j \neq i} b_{ij} \geq A_i^t \quad 3.5$$

$$\sum_{i \in hostk}^{n} A_i^t \leq B_k \quad 3.6$$

$$\forall i \sum_{j \neq i} G_{ij} \leq A_i^t \quad 3.7$$

Ti represents the traffic sending request of VPN i, based on the Model above,

$$T_i = \sum_{j \neq i} b_{ij} \quad 3.8$$

a) Modified Dynamic Hose Model on Virtual Private Network to Enhance Resource Management Using MATLAB/SIMULINK

The Modified Dynamic Hose Model was utilized in conjunction with a MATLAB program and SIMULINK model to conduct a simulated analysis of resource management in a virtual private network while accounting for different traffic rates. The monitoring method will identify VPN congestion using the dynamic host model approach. MATLAB is an interactive environment and high-level technical computing language used for numerical calculation, data visualization, analysis, and algorithm creation. It contains a large number of power simulation libraries. Matrix laboratory is what the word MATLAB stands for. A software package for high-speed numerical computing and visualization is called MATLAB, and it was created by MathWorks Inc. MATLAB is the best program for scientific researchers because of its outstanding graphics, flexibility, dependability, and analysis capabilities. Hundreds of dependable and precise built-in mathematical functions are available in an interactive environment called MATLAB. Numerous mathematical issues, such as matrix algebra, complex arithmetic, linear systems, differential equations, signal processing, optimization, nonlinear systems, and many other kinds of scientific computations, can be resolved by using these functions. For specialized applications like signal processing, fuzzy logic, control systems design, system identification, statistics, neural networks, and symbolic computations, among others, there are a number of extra toolboxes available. The extremely potent Simulink application has improved MATLAB.

A software program called Simulink is used to model, simulate, and analyze dynamical systems. Both linear and nonlinear systems can be described in sampling time, continuous time, or a combination of the two. Systems can also include multiple rates of sampling or updating, or multi-rate components. Simulink offers a graphical user interface (GUI) for modeling that allows users to create block diagram models by dragging and dropping mouse points. You can draw the models using this interface in the same way as you would with a pencil and paper, or as most textbooks show them. A vast block library of sources, sinks, linear and nonlinear components, and connections is included with Simulink.

The input box in the simulation generates data packets. The packet size is represented by properties in each piece of data. Next, the data packets are sent through the data network. The MATLAB/Simulink Model for Resource Management in Virtual Private Networks is shown in Figure 3.8. The principal components included traffic source modules, network sink, logic module (if different), parameter input box, display box, ingress committed rate, dynamic hose model, resource assignment, and scope.

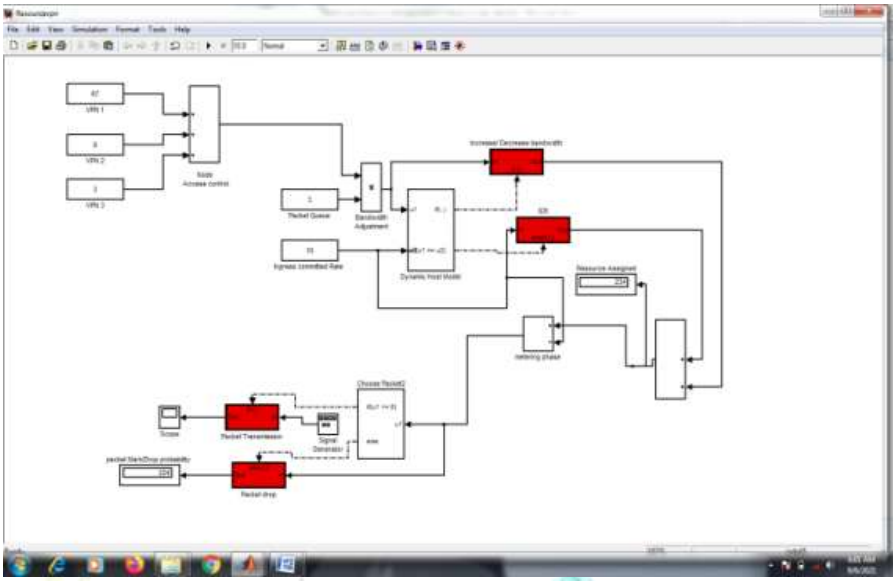

Figure 4: MATLAB Model of Modified Dynamic Hose Model on Virtual Private Network

Three VPNS are used in this work's model to fill it with traffic; the system is scalable and capable of supporting other VPNS. Three traffic sources—voice, video, and best-effort packets—are used by each VPN.

IV. RESULTS AND ANALYSIS

1) Parameters for Simulation

MATLAB - SIMULINK was used to carry out the simulation. We determined the Mark/Drop probability for the Hose Model Algorithm in the simulation, and the same process was followed for the Modified Hose Model Algorithm. The proposed algorithm is compared with the conventional (existing) algorithm in the simulation. Table 1 displays the simulation parameters that were used.

Table 1: Simulation Parameters

Parameter	Value
Protocols	UDP
Number of VPN	3
Data rate	10 Mb, 50 Mb, 100 Mb
Packet Size	512kb Bytes, 1024 Bytes, 2048 Bytes
Simulation Time	13s
Traffic Pattern	CBR, FTP
Ingress committed rate	10

2) Simulation Results

a) Network Characterization Result

Since all of the VPN servers we used had rather high bandwidths—roughly 20 Mbps—we employed large files for testing workloads in the network characterization. File sizes of at least 15 MB were chosen to give the

network traffic time to settle and gather more accurate data. The throughput study performed on each of the client VPN servers separately is displayed in the next two tables.

Table 2: Throughput Testing servers 1

Number of Users	Throughput (KB/sec)	Per User (KB/sec)	Bandwidth utilization
1	3638.19	3638.19	16.12
2	5769.92	2884.96	25.57
4	12042.24	3010.56	53.36
8	19288.32	2411.04	85.48
10	20722.33	2072.23	91.83
12	22564.14	1880.34	100
16	22355.36	1397.21	99.07

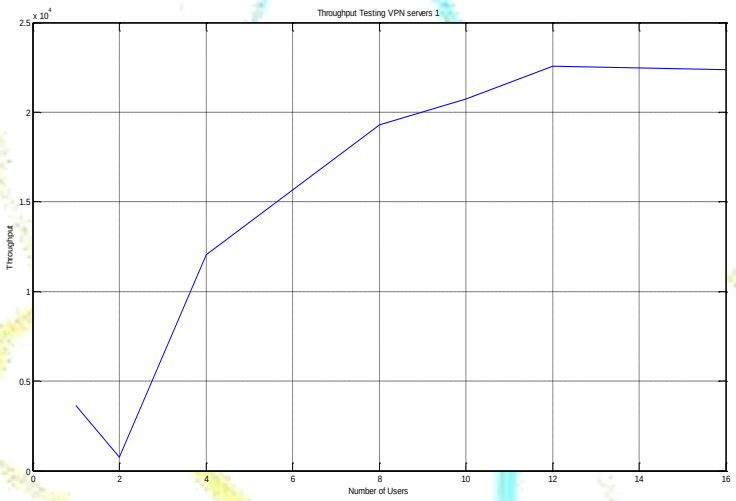

Figure 7: Number of Users versus Throughput Testing VPN servers 1

Figure 7 shows that the Throughput increases with an increase in the number of Users showing that more Packets are transmitted and delivered on the Network.

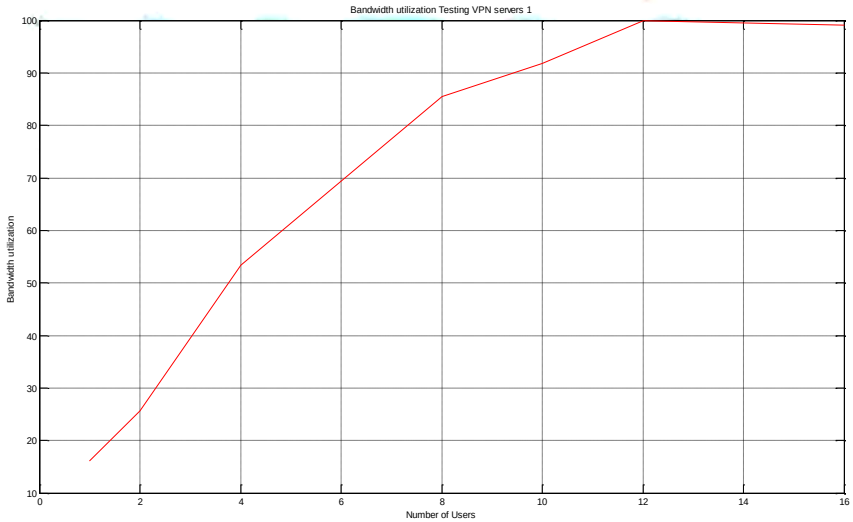

Figure 8: Number of Users versus Bandwidth Utilization Testing VPN servers 1

Figure 8 illustrates how, as the number of users increases, the bandwidth allotted to each user decreases. However, we see that there are about 12 Users per Client when the maximum bandwidth is approaching. Another thing we've noticed is that when we load the client with more users, the bandwidth use goes up. This is a flaw in the files system because it ought to always strive to maximize bandwidth use.

Table 3: Throughput Testing VPN servers 2

Number of Users	Throughput (KB/sec)	Per User (KB/sec)	Bandwidth utilization
1	3723.05	3723.05	14.72
2	5739.89	2869.945	22.70
4	13883.89	3470.97	54.92
8	24454.57	3056.82	96.74
10	23722.33	2372.23	93.84
12	25278.57	2106.54	100
16	22355.36	1397.21	88.43

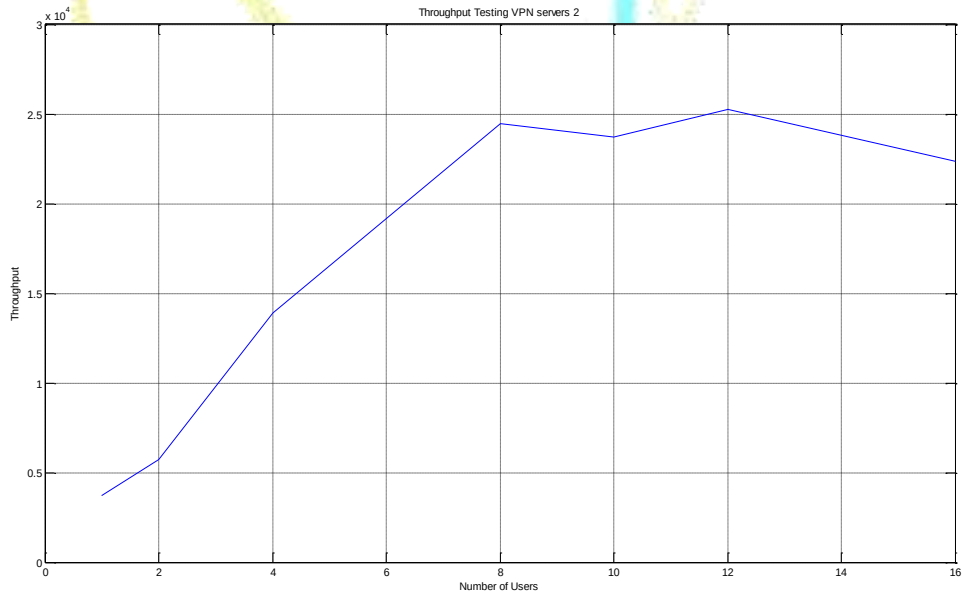

Figure 9: Number of Users versus Throughput Testing VPN servers 2

Figure 9 illustrates how the throughput rises as the number of users increases, indicating that more packets are sent over the network and received. This demonstrates that the throughput characteristics of VPN servers 1 and 2 are comparable, and they may be used with any VPN network.

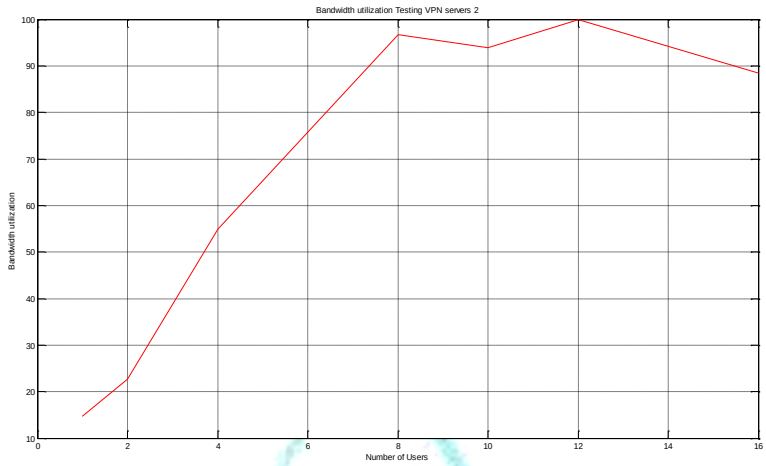

Figure 10: Number of Users versus Bandwidth Utilization Testing VPN servers 2

The bandwidth allotted to each user decreases as the number of users rises, as Figure 10 illustrates. However, we see that there are about 12 Users per Client when the maximum bandwidth is approaching. Another thing we've noticed is that when we load the client with more users, the bandwidth use goes up. This is a flaw in the files system because it ought to always strive to maximize bandwidth use. Additionally, this demonstrates that VPN servers 1 and 2 share comparable bandwidth use characteristics and can be used with any VPN network.

Various elements of packet size and transfer speeds are used to run different tests in turn. Table 4 displays the simulation results with CBR running on the UDP protocol.

Table 4: CBR Throughput for different Packet sizes at the CBR transfer rate: 10 Mb, 100 Mb and 1000 Mb

Time	**Packet size: 512 Bytes**	**Packet size: 1024 Bytes**	**Packet size: 2048 Bytes**
2	0	0	0
2.5	0.008	0.016	0.033
3	0.008	0.016	0.033
3.5	0.008	0.016	0.033
4	0.008	0.016	0.033
4.5	0.008	0.016	0.033
5	0.008	0.016	0.033
5.5	0.008	0.016	0.033
6	0.008	0.016	0.033
6.5	0.008	0.016	0.033
7	0.008	0.016	0.033

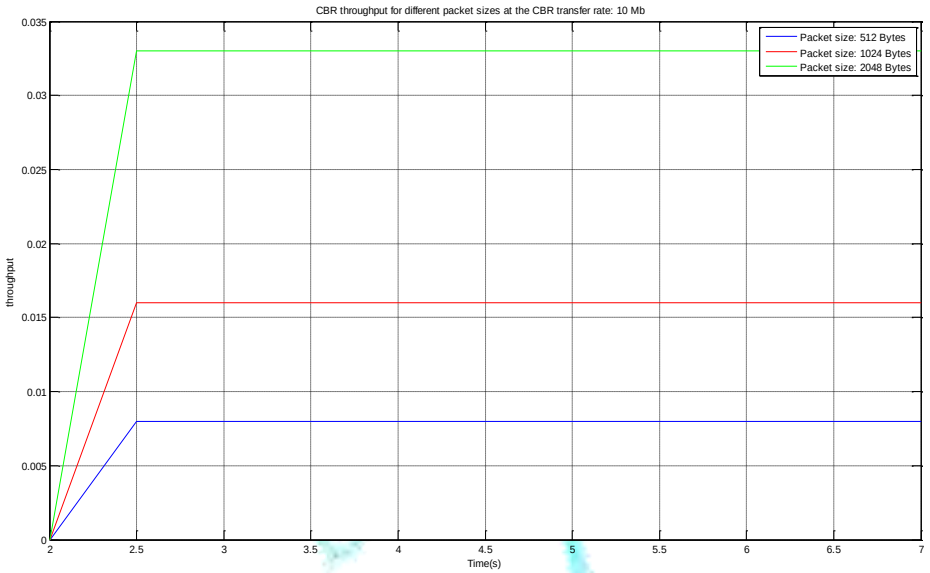

Figure 11: Time versus CBR Throughput for different Packet sizes at the CBR Transfer rate: 10 Mb, 100 Mb, and 1000 Mb

Packet size and transfer rate are the two criteria used in the characterization of VPN throughput. Figure 4.5 illustrates how packet size impacts VPN throughput. Transfer rate has no effect on throughput in a VPN since all three transfer speeds offer the same throughput.

The FTP protocol is tested with various window and packet sizes while utilizing the TCP protocol; the experimental findings are shown in Tables 5, 6, and 7.

Table 5: FTP Throughput with FTP Window size: 10Kb

Time	**Packet size: 512 Bytes**	**Packet size: 1024 Bytes**	**Packet size: 2048 Bytes**
8	0	0	0
8.5	0.48	0.92	1.8
9	0.79	1.53	3.01
9.5	0.71	1.36	2.67
10	0.71	1.36	2.67
10.5	0.79	1.36	2.67
11	0.71	1.53	3.01
11.5	0.71	1.36	2.67
12	0.79	1.36	2.67
12.5	0.71	1.53	2.67
13	0.71	1.36	3.01

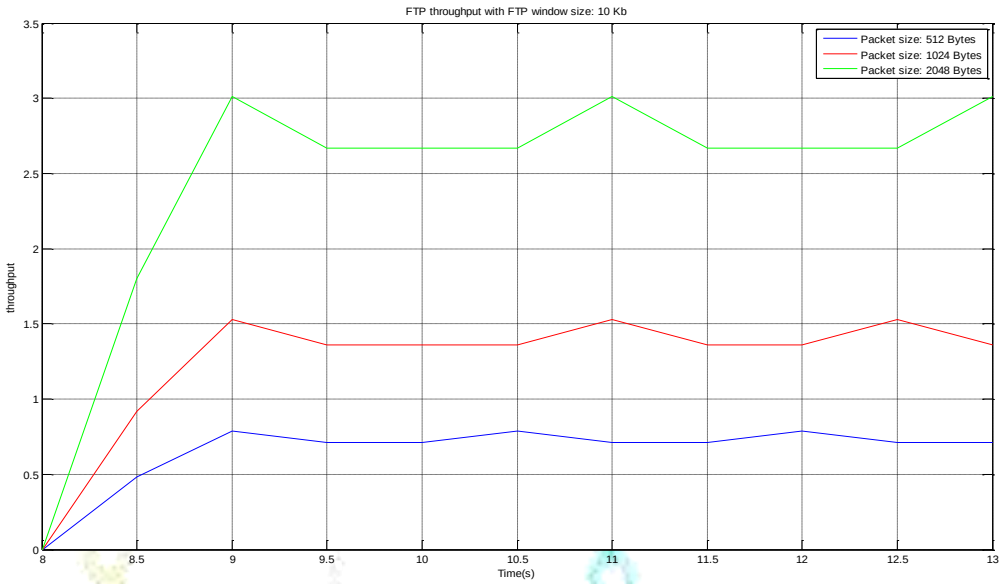

Figure 12: Time versus FTP throughput with FTP window size: 10Kb

Figure 12 demonstrated how, with packet sizes of 512 bytes, 1024 bytes, and 2048 bytes, the window size is employed to trigger outcomes in turn. This demonstrates how variations in packet and window sizes impact a VPN's throughput.

Table 6: FTP Throughput with FTP window size: 50Kb

Time	**Packet size: 512 Bytes**	**Packet size: 1024 Bytes**	**Packet size: 2048 Bytes**
8	0	0	0
8.5	1.43	2.76	5.41
9	3.97	7.66	13.73
9.5	3.53	6.81	14.67
10	3.53	6.81	13.36
10.5	3.97	6.81	13.36
11	3.53	7.66	14.37
11.5	3.53	6.81	14.03
12	3.65	6.81	13.36
12.5	3.85	7.66	13.36
13	3.53	6.81	15

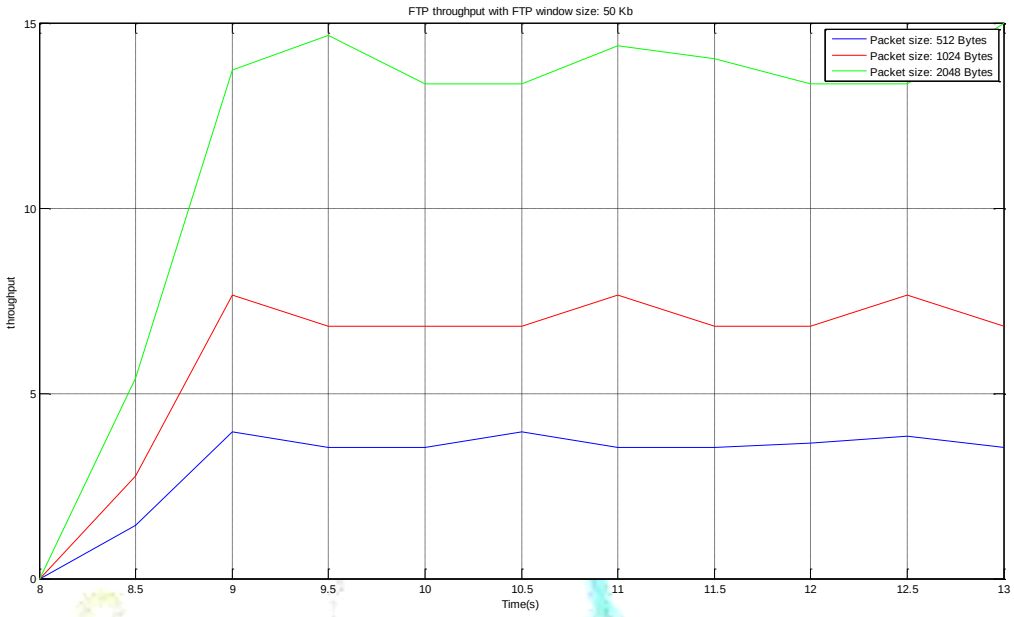

Figure 13: Time versus FTP Throughput with FTP Window size: 50Kb

Figure 13 demonstrated how the window size is utilized to alternately trigger results with packet sizes of 512 bytes, 1024 bytes, and 2048 bytes. This demonstrates how changes in Window and Packet sizes impact a VPN's Throughput, as the Throughput generated increased when Window size was increased from 10kb to 50kb. In this case, the greatest throughput achieved was 15, compared to 3.01 when a window size of 10Kb was utilized. This results in an improvement of 11.99, or 79.93%.

Table 7: FTP Throughput with FTP Window size: 100Kb

Time	Packet size: 512 Bytes	Packet size: 1024 Bytes	Packet size: 2048 Bytes
8	0	0	0
8.5	1.997	3.85	7.55
9	7.95	14.78	27.09
9.5	7.07	14.16	29.7
10	7.07	13.62	26.73
10.5	7.9	13.62	26.73
11	7.11	15.32	27.73
11.5	7.07	13.62	29.06
12	7.19	13.62	26.73
12.5	7.83	15.32	26.73
13	7.07	13.62	28.3

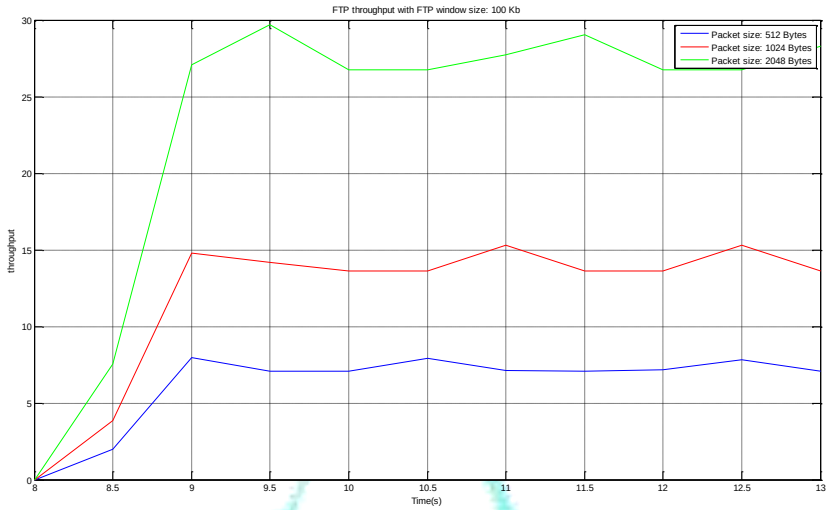

Figure 14: Time versus FTP Throughput with FTP Window size: 100Kb

Figure 14 demonstrated how the Window size—which uses packet sizes of 512 bytes, 1024 bytes, and 2048 bytes—is utilized to trigger results in turn. This demonstrates how changes in Window and Packet sizes impact a VPN's Throughput, as the Throughput generated increased when Window size was increased from 50 kb to 100 kb. In this case, the greatest throughput produced was 28.3, compared to 15 when the window size of 50Kb was utilized. This represents a 13.3 or 47% improvement.

b) Packets Mark/Drop Probability

With MATLAB, the data gathered from the Model was simulated. In order to provide efficient data rate throughput in the network operation, handle varying traffic rates, and improve the mark/drop probability with different values of congestion window, the simulation results are presented in Tables 8, 9 and 10 and Figures 15, 16 and 17. These results show the variations in mark/drop probability with different values of congestion window for the Hose Model Algorithm and Dynamic Hose Model as applied to Model Resource Management and Quality of Service control in VPN using Modified Dynamic Hose Model.

Table 8: Simulated Data of Mark/Drop probability with different values of Congestion Window using Hose Model Algorithm

Traffic congestion window	**mark/drop probability**
2	2.4448
4	1.5758
8	0.8717
16	0.4512
32	0.2283
64	0.1147
128	0.0575

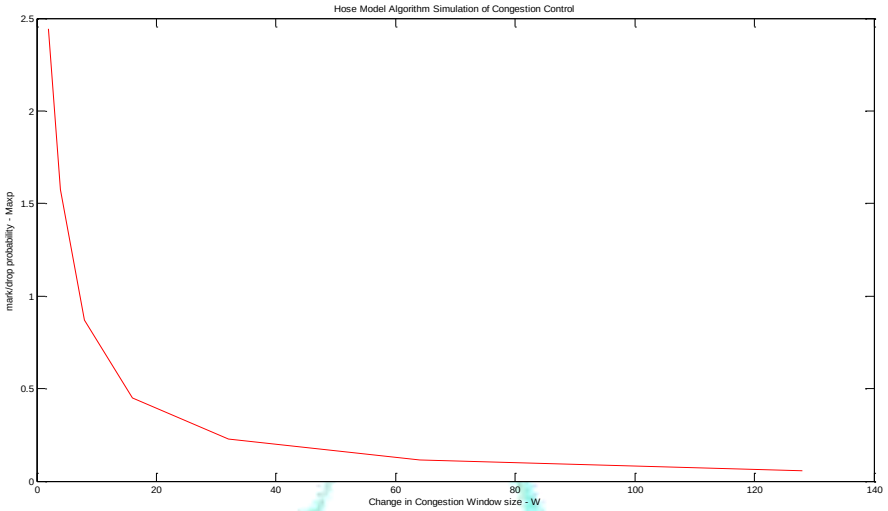

Figure 15: Variations in mark/drop probability with different values of Congestion window W Packets, for Hose Model Algorithm

An analysis of the Hose Model Algorithm's simulation of the variations in mark/drop probability with various congestion window W values, or packets, as illustrated in Figure 15, reveals that the variations in mark/drop probability with various window sizes W exhibit a dropping characteristic. This suggests that increasing the window size will decrease the mark/drop probability as illustrated in Figure 15.

Table 9: Simulated Data of Mark/Drop probability with different values of Congestion Window using Modified Dynamic Hose Model

Traffic congestion window	**mark/drop probability**
2	1.6284
4	1.0495
8	0.5806
16	0.3005
32	0.1521
64	0.0764
128	0.0383

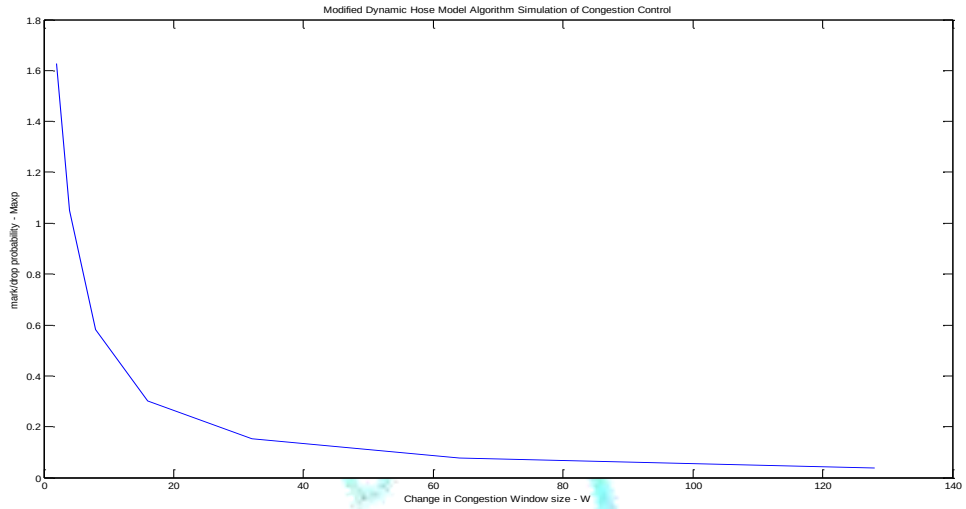

Figure 16: Variations in Mark/Drop probability with different values of Congestion window W, Packets for Modified Dynamic Hose Model

According to Figure 16, which simulates variations in mark/drop probability with various values of the congestion window W and packets for the modified dynamic hose model, there are dropping characteristics associated with these variations in mark/drop probability with varying window sizes W. This indicates that while an increase in window size will decrease mark/drop probability, the quality of the drop was enhanced by differentiated service. By implementing sophisticated QoS features as conditioning, marking, and classification utilizing the DiffServ into a restricted number of traffic aggregates or classes exclusively at the Edge Nodes, the Modified Dynamic Hose Model provides scalability.

Using IP DiffServ QoS methods, all traffic conditioning and dropping is intelligently managed at the Network Layer in the Core Routers. Scheduling and queuing control mechanisms are applied to the traffic classes based on the field marking.

Table 10: Simulated Data of Mark/Drop probability with different values of Congestion window using Hose Model and Modified Dynamic Hose Model Algorithm

Traffic Congestion Window	**Mark/Drop probability**	
	Hose Model	**Modified Dynamic Hose Model**
2	2.4448	1.6284
4	1.5758	1.0495
8	0.8717	0.5806
16	0.4512	0.3005
32	0.2283	0.1521
64	0.1147	0.0764
128	0.0575	0.0383

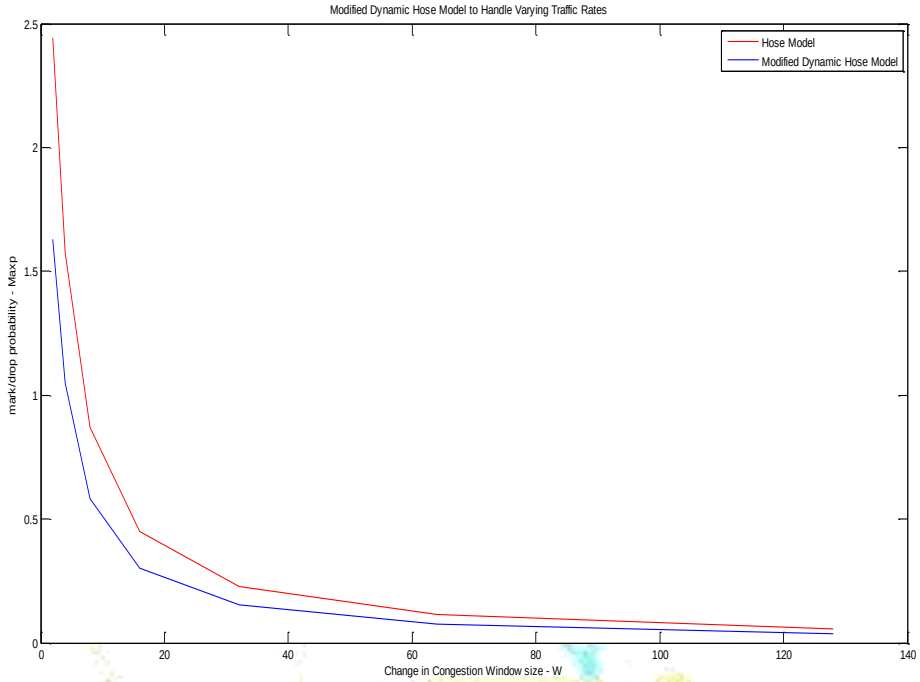

Figure 17: Variations in Mark/Drop probability with different values of Congestion Window W, Packets for Hose Model and Modified Dynamic Hose Model Algorithm Compared

The Hose Model Algorithm has more dropping characteristics for variations in Mark/Drop probability with different window sizes W than the Modified Dynamic Hose Model Algorithm, according to a simulation of the variations in Mark/Drop probability with different values of Congestion Window W, Packets for Hose Model, and Hose Model Algorithm. As a result, the drop preferences for packets that are in and out of profiles will differ. Next, the Modified Dynamic Hose Model Algorithm decides whether to queue or drop a specific packet. To prevent out-of-order delivery, all packets that are not dropped regardless of whether they are in or out of profile are added to a single queue.

c) Packet Delivery Ratio

Table 11: Comparison of Sent (ICR) and Received (ECR) Packets for Existing and Proposed Models

Number of Packet Sent	**Number of Packet Received with Existing Model**	**Number of Packets Received with Proposed Model**
50	36 (0.72%)	46 (0.92%)

The Table.11 represents the Packets delivery ratio which is calculated as the total number of Packet received successfully at the final destination and total number of Packets sent.

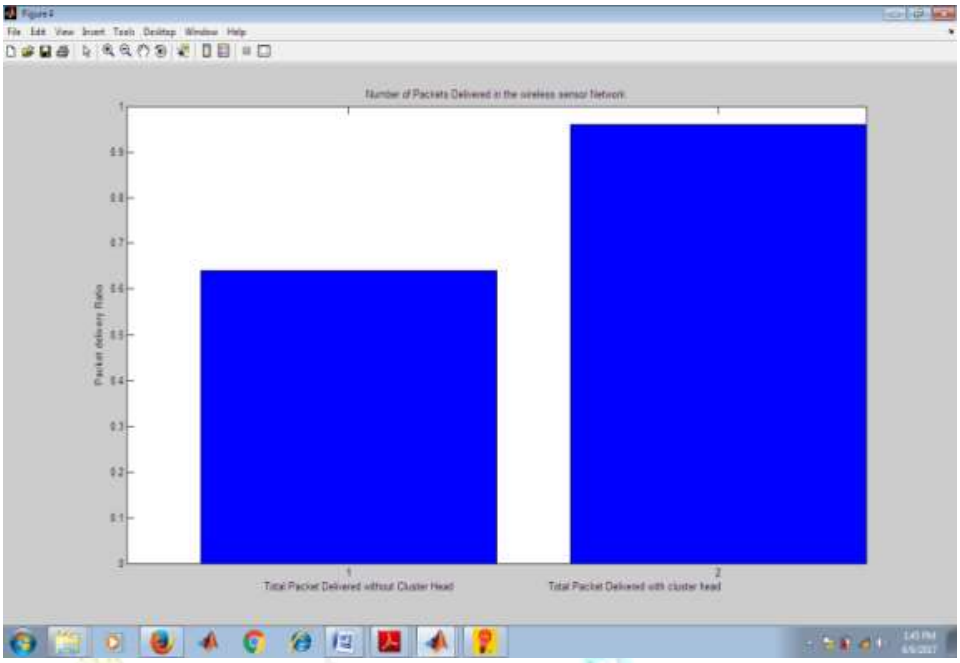

Figure 18: Comparison for Packets delivery Ratio for Existing Pipe Hose and the Proposed Modified Dynamic Hose Technique.

Figure 18 showed the variation in the Packets delivery ratio of the Existing and Proposed Techniques and Comparison of both in terms of Packets delivery performance. The Modified Dynamic Hose Model posted 92% Packet delivery ratio as against the Existing Model that posted 72% Packet delivery ratio.

V. CONCLUSION

The Modified Dynamic Hose Model was simulated in this paper in order to enhance resource management in virtual private networks. To acquire the results of the created parameters for each simulation time employed, the necessary parameters were supplied to the MATLAB workspace. Figures 7 and 8 demonstrate how more packets are sent and received over the network and how throughput rises as the number of users increases. This demonstrates that the throughput characteristics of VPN servers 1 and 2 are comparable, and it can be applied to any VPN network. The bandwidth allotted to each user decreases as the number of users rises, as seen in Figures 8 and 10. However, we see that there are about 12 Users per Client when the maximum bandwidth is approaching. Another finding is that when we load the client with more users, the bandwidth use goes up. The files system should always try to maximize bandwidth use, hence this is a disadvantage. Additionally, this demonstrates that VPN servers 1 and 2 share comparable bandwidth utilization characteristics, and it can be applied to any VPN network. Various elements of packet size and transfer rates are used to run different tests in turn. Table 4 displays the simulation results with CBR running on UDP protocol. Packet size and transfer rate are the two criteria used in the characterization of VPN throughput. Figure 11 illustrates how the three transfer rates offer the same throughput while the packet size influences the VPN throughput. Thus, in a VPN, transfer rate has no bearing on throughput. Tests are conducted with various Window and Packet sizes with FTP through the TCP protocol. Tables 5, 6, and 7 display the outcomes of the experiment. Figure 12 demonstrated how the Window size is utilized with Packet sizes of 512 bytes, 1024 bytes, and 2048 bytes to trigger outcomes in turn. This demonstrates how changes in Window and Packet sizes impact a VPN's Throughput, as the Throughput generated increased when Window size was increased from 10 kb to 50 kb. In this case, the greatest throughput generated was 15, compared to 3.01 obtained with a window size of 10Kb. This results in an improvement of 79.93%. Figure 13 demonstrated how, with packet sizes of 512 bytes, 1024 bytes, and 2048 bytes, the Window size is used to trigger outcomes in turn. This demonstrates how changes in Window and Packet sizes impact a VPN's Throughput, as the Throughput generated increased when Window size was increased from 50 kb to 100 kb. In this case, the greatest throughput produced was 28.3, compared to 15 when the window size of 50Kb was utilized. This represents a 47% improvement. In order to provide efficient data rate throughput in the network operation, handle varying traffic rates, and improve the Mark/Drop probability with different values of Congestion Window, the simulation results are presented in Tables 8, 9 and

10, and Figures 14, 15, and 16. These figures show the variations in Mark/Drop probability with different values of Congestion Window for Hose Model Algorithm and Dynamic Hose Model as applied to Model Resource Management and Quality of Service Control in VPN using Modified Dynamic Hose Model. The results of a simulation comparing the Packets for Hose Model and Modified Dynamic Hose Model Algorithm with varying Congestion Window W values indicate that the Hose Model Algorithm exhibits more dropping characteristics when it comes to Mark/Drop probability variations with varying Window sizes W, as illustrated in Figure 17. For this reason, the Drop preference for in-profile and out-of-profile packets will differ. Next, the Modified Dynamic Hose Model Algorithm decides whether to queue or drop a specific packet. To prevent out-of-order delivery, all packets that are not dropped—regardless of whether they are in or out of profile—are added to a single queue. Figure 18, Comparison for Packets delivery Ratio for Existing Pipe Hose and the Proposed Modified Dynamic Hose Technique, showed the variation in the Packets delivery ratio of the Existing and Proposed Techniques and Comparison of both in terms of Packets delivery performance. The Modified Dynamic Hose Model posted 92% Packet delivery ratio as against the Existing Model that posted 72% Packet delivery ratio.

REFERENCES

[1] Fotedar, S., Gerla, M., Crocetti, P. and Fratta, L., ATM Virtual Private Networks, Communications of the ACM, vol. 38, 2015, pp. 101–109.

[2] ISO/IEC JTC 1/SC 29/WG 11, Information technology – coding of audio – visual objects, part 1: systems, part 2: visual, part 3: audio, FCD 14496.

[3] Paxson, V., End-to-End Internet Packet Dynamics, In Proc. of ACM SIGCOMM, 2017.

[4] Report ITU-R M.2244, Standards in Isolation between Antennas of IMT Base Station in the Land Mobile Service, 2011.

[5] Guerin, R. and Peris, V., Quality-of-service in packet networks: basic mechanisms and directions, Computer Networks and ISDN, vol. 31, no. 3, pp. 169–179, 2010.

[6] Okorogu V.N, Ogbodo E.U., Okafor, C.S., Obioma, P.C., Development of an Energy-Efficient Routing Algorithm for Long Distant Wireless Sensor Network Infrastructure, Journal of Electronics and Communication Systems, Volume-8, Issue-2, pp.43-59, 2023.

[7] N C Maduka, C S Okafor and E I Archibong, Investigation of quality of service (QoS) of GSM network providers at Federal University, Gusau, Zamfara State, Nigerian, Journal of Physics, 30(2), 77-85, Available at: https://www.researchgate.net/publication/36 7238468_Investigation_of_Quality_of_service_QoS_of_GSM_Network_Providers_at_Federal_University_Gusau_Zamfara_State, 2021.

[8] Okorogu V.N, Ogbodo E.U., Okafor, C.S., Obioma, P.C., Optimization of Data Throughput for Improved Traffic Management in a Data Switched Network, Journal of Electronics and Communication Systems, Volume-8, Issue-2, pp.10-29, 2023.

[9] Roch, A. G. and Kumar, N. S., Delay and Throughput Performance of Speeded-Up Input-Queueing Packet Switches. IEEE Journal of Selected Areas in Communications, vol. 5, no. 8, pp. 1264–1273, 2017.

[10] Satish, R. and Ramakrishnan, .K.K., Resource Management for Virtual Private Networks. IP VPNs, In Proc. of IMC 2017, pp. 342–355.

[11] Christian, M., Dotaro, E., Papadimitriou, D., A Practical Approach to VPN Resource Management using a Dynamic Hose Model. 2016 2nd Conference on Next Generation Internet Design and Engineering, Valencia, 2016, 147-153.

[12] Lim, L.K., Gao, J., Ng, T.S.E., Chandra, P.R., Steenkiste, P. and Zhang, H., Customizable Virtual Private Network Service with QoS. Computer Networks, 36, 137-151, 2017.

[13] Wei, D., Ansari, N., Implementing Fair Bandwidth Allocation Schemes in Hose-modelled VPN. IEE Proceedings-Communications, 151, 521-528, 2014. ttps://doi.org/10.1049/ip-com:20040840

[14] Alpar, J., Istvan, S. and Aron, S. (2013) On Bandwidth Efficiency of the Hose Re-source Management Model in Virtual Private Networks. 22nd Annual Joint Conference of the IEEE Computer and Communications, 1, 386-395, 2013.

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

[16] Balmer, R., Baumgartner, F., Braun, T., Günter, M., Khali, I., Virtual Private Network and Quality of Service Management Implementation. RFC 2475, 2013.